RADC-TR-72-171
Final Report
July 1972

DATA MANAGEMENT SYSTEM TEST METHODOLOGY DEVELOPMENT

PRC/Information Sciences Company

Approved for public release;
distribution unlimited.

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) PRC/Information Sciences Company 7600 Old Springhouse Road McLean, Virginia 22101	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE

DATA MANAGEMENT SYSTEM TEST METHODOLOGY DEVELOPMENT

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report April 1971 - April 1972

5. AUTHOR(S) (First name, middle initial, last name)

William W. Ferguson
Frank M. McGoldrick, et al

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1972	278	3

8a. CONTRACT OR GRANT NO F30602-71-C-0252 Job Order No. 55500000	9a. ORIGINATOR'S REPORT NUMBER(S) R-1604 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-72-171

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Rome Air Development Center (ISIM) Griffiss Air Force Base, New York 13440

13. ABSTRACT

This document proposes a methodology to be employed in the testing of data management systems and submits some recommendations for the continued development of a DMS Test Methodology. The intent of this document was first to characterize a data management system by identifying the various attributes that should comprise a DMS and summarize the techniques that can be employed in implementing these capabilities. Secondly, the standard test techniques that can be used to measure the capabilities of the aforementioned attributes were examined and, based on the conclusions derived from this analysis, a DMS Test Methodology was proposed. To assist in methodology utilization, a correlation between particular attributes and specific measurement techniques was drawn and scenarios were written to illustrate how the methodology would be employed in the solution of some typical DMS measurement problems. Finally, it was concluded that analysis, benchmark programs, and software monitors were the most useful test techniques available and warrant additional development.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Data Management System						
DMS Testing						
DMS Evaluation						
Software Testing						
Test Methodology						
DMS Characteristics						
Software Test Techniques						
Benchmarks						
Monitors						
Modeling						
Simulation						
Test Scenarios						
DMS Characteristic Test Pairing						

1B

UNCLASSIFIED
Security Classification

DATA MANAGEMENT SYSTEM TEST METHODOLOGY DEVELOPMENT

William H. Ferguson
Frank [illegible]. McGoldrick
et al

PRC/Information Sciences Company

Approved for public release;
distribution unlimited.

FOREWORD

This document is the final technical report on work performed under contract F30602-71-C-0252, Job Order Number 55500000, for Rome Air Development Center, Griffiss Air Force Base, New York, by Planning Research Corporation, Information Sciences Company, 7600 Old Springhouse Road, McLean, Virginia.

The purpose of this contract was to characterize a data management system, subsequent to which a methodology was to be developed to test same, using standard test techniques which already exist or techniques which have been identified for new development. The work was performed between April 1971 and April 1972. Significant assistance and guidance was received from RADC program monitor Francis P. Sliwa of the Information Management Sciences Section, Information Sciences Division (ISIM).

The principal investigator for the project was William W. Ferguson with significant assistance from Frank M. McGoldrick, K. Randa Knorr, K. Richard Stewart, and Andrew A. Makowka.

This report is numbered R-1604 for PRC/ISC internal control.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

Approved: FRANCIS P. SLIWA
Info Mg. Sciences Section
Info Processing Branch

Approved: WILLIAM P. BETHKE
Chief, Information Sciences Div.

FOR THE COMMANDER: Fred I. Diamond

FRED I. DIAMOND
Acting Chief, Plans Office

ABSTRACT

This document proposes a methodology to be employed in the testing of data management systems and submits some recommendations for the continued development of a DMS Test Methodology. The intent of this document was first to characterize a data management system by identifying the various attributes that should comprise a DMS and summarize the techniques that can be employed in implementing these capabilities. Secondly, the standard test techniques that can be used to measure the capabilities of the aforementioned attributes were examined and, based on the conclusions derived from this analysis, a DMS Test Methodology was proposed. To assist in methodology utilization, a correlation between particular attributes and specific measurement techniques was drawn and scenarios were written to illustrate how the methodology would be employed in the solution of some typical DMS measurement problems. Finally, it was concluded that analysis, benchmark programs and software monitors were the most useful test techniques available and warrant additional development.

TECHNICAL EVALUATION

The technical effort reported herein is a part of a continuing program at RADC to develop a Data Management System Test and Evaluation Methodology. This report reflects the first year of effort which bounded the DMS testing problem by listing all the DMS characteristics which might be tested and compiling the currently used testing techniques. The recommended application of test techniques to particular DMS characteristics is based on the author's experiences and from a thorough literature search. Additional experience was gained from a separate RADC contract entitled "DMS Testing and Methodology Validation".

It is expected that the recommendations and conclusions in the report will be modified as further developments or refinements of test techniques are made. The most important contribution of this effort is that it provides a checklist of DMS features and test methods with a stepwise procedure which a DMS tester can follow to improve the tester's level of confidence. This should also decrease the time to plan a test program which should provide desired test results with the least effort.

Francis P. Sliwa

FRANCIS P. SLIWA
Technical Evaluator

TABLE OF CONTENTS

SECTION I

INTRODUCTION

1. INTRODUCTION

This document describes the results of a study of the means of testing generalized software systems, particularly Data Management Systems (DMS). This paper describes the functional characteristics of these systems; the attributes, methods and results of testing; provides a methodology for selecting pertinent test mechanisms by pairing DMS characteristics with identified tests; illustrates technique utilization by means of test scenarios; and, finally, recommends areas for which appropriate test technology is weak or missing.

2. DMS OVERVIEW

The DMS technology has been in a state of continuing development since the late 1950's. Early efforts were directed primarily at military problems of storing, maintaining, retrieving, manipulating, and presenting results of formatted data stored in massive files. Subsequent efforts have been directed at improving solutions to military problems and developing generalized systems for broader applications. Corporate requirements for information handling--not unlike those of the military--have spawned additional developments of generalized DMSs.

It is appropriate at this time to review the concept of what a DMS actually is, the components that go toward constituting a DMS and its fundamental capabilities.

A data management system is best described as an independent set of software that facilitates the generation, maintenance, query and analysis of a data base. It is considered generalized when it permits the manipulation of newly defined files and data without requiring the modification of existing programs. Its major components are:

- query language
- data description language
- file maintenance language
- file structuring capability
- data access manipulation and output capability

The degree to which each data management system possesses these capabilities will vary from one system to another, but, in general most systems support the following functions:

- Enables data references by name rather than physical location
- Supports the expression of logical relationships among data elements

- Supports data operations such as definition, storage, maintenance, retrieval, and presentation
- Facilitates newly defined files and data operations

It is apparent that the definition of a generalized DMS is sufficiently broad to cover an extremely wide variety of systems. In fact, a cursory examination of the literature reveals more than 75 systems that qualify. This proliferation of DMSs presents a difficult problem to the systems developer, who must determine the most appropriate method for solving sets of application problems.

While the purpose and use of data management systems may make their selection and measurement appear relatively simple, a detailed investigation of DMS elements and functions reveals a complex software problem. In testing a DMS, a potential user must specify the major DMS elements and their effects on his requirements. In this regard, the major elements of a DMS can be listed as:

- Languages to be used, including generalized routines to be provided.
- Libraries to be used
- File organization and the method of accomplishing content retrieval
- The environment in which the system will operate
- The scope of the system.

There are many factors that a data management system user must consider. He must consider a variety of languages, libraries, file organizations, operating environments and possible system scopes. The number of possible combinations of system designs is huge.

Obviously, the system user must choose the pertinent DMS features he wishes to test. If he attempts to include more and more features the complexity of the testing process increases and the test may become unfeasible. On the other hand, if the test personnel restrict the elements of the systems to be measured to simplify the task, the test may become useless.

In the early days of DMS testing, most users imposed a rather strict set of qualifications on a system in order to render the system operational in a user environment. Because of this attitude later DMS users rejected these conclusions on the grounds of their cursory study and limited approaches. Furthermore, the management rationale for utilizing a DMS has undergone considerable change. Data processing management is now more concerned with its most limited resources; analyst and programmer times. Moreover computer processing costs are coming down rapidly. Data processing management must now insure that a DMS is not being rejected by test techniques that stress machine efficiencies at the expense of other resources--e.g., human resources.

Data management systems represent a major trend in the data processing field. Development of these systems has been somewhat slower than expected because of their complex design and the resulting complex systems evaluations. Once a test methodology is found that can be related to several different types of measurement problems, e.g., comparing two data management systems, testing a DMS as a stand-alone system, or testing a DMS performing a special application, then data processing management will be more inclined to use such a tool.

SECTION II

DMS ATTRIBUTES

This section presents the attributes that comprise a generalized data management system. These attributes are divided into the following major categories:

- o Data structure and definition.
- o Data manipulation functions.
- o DMS system control.
- o Host environment interrelationships.

Each of these major categories is preceded by a brief introduction, following which, the various attributes are described in detail.

1. Data Structure and Definition

a. Logical Structure

All generalized data base management systems provide a conceptual method of organizing data into meaningful structures. Made available to the system user is a set of logically related categories which he can apply to his data in order to create files of validly organized information. However, he does not necessarily have to apply them to the organization of his data in secondary storage. Through this provision, a user can interact with data in terms independent of the manner in which the data are physically stored.

The data categories provided range from the smallest designated element of structure to which data can be assigned to the entire data base itself. In this hierarchy of structural categories, the most elementary can be given such dimensional attributes as length, identification and security and be grouped together by the user, in accordance with the way in which he wishes to organize his data, to build the next higher structure which can in turn be assigned certain attributes and be organized with others of its kind as constituents of the next structure, and so on. After the structural organization has been determined, the data values themselves may be manipulated using system provided functions to populate the categories and thus create the data base, or to update the categories if they have been previously populated. This section describes data structure classes which systems make available to the user for the creation of a data base through user specified creation and update processes. The description outlines the set of attributes for each class which distinguishes it from other classes, and the values these attributes may assume.

(1) Structure Classes

Five structure classes can be postulated as common to nearly all data management systems even though system terminology may vary. These classes and the terms used in this document to describe them are as follows:

Preceding page blank

(a) Item

A single elementary data entity containing no logical substructure and from which all other types of structures are ultimately composed. The principal attribute of the item is the value. Other attributes might include identification, type, security, and value existence indicators.

(b) Group

A set of items and possibly other groups. Groups can be simple or compound, repeating or non-repeating. A group composed solely of items is called a simple group; a group composed of a set of items and a set of related groups is called a compound group. A set of item values for all items comprising the group is called an instance of the group. A group is the lowest level of data structure concerned in the logical organization of a data base; it can maintain at once three different relationships with other groups: parent (superior), dependent (subordinate), and peer. Within a compound group, groups can be manipulated to establish a hierarchic organization of these relationships. (The term hierarchic implies that each parent group instance can be optionally paired with one or more dependent group instance.) The kinds of hierarchical relationships that can be organized by the user depend upon attributes of group composition; i.e., the way in which single groups are arranged to compose another structure. If the system provides a group relation structure, however, logical organization will depend upon the attributes of this type of structure (see following paragraph). Other possible attributes of the group in addition to composition are type, identification and security.

(c) Group Relation

The logical relation or mapping between two sets of groups, the first set being the parent groups and the second set being the dependent groups. A group relation has a set of attributes of its own. The relationship facility provided is equivalent to the hierarchic group relationship facility provided by a compound group, except for the following differences:

- In a compound group, a group may be subordinated to one parent group only; in a group relation, a group may be subordinated to many parent groups. Such a relationship is often termed non-hierarchic.

- In a compound group, the principal items (the immediate constituents of a compound group to which a group is subordinated) do not have a collective name. In a group relation, each set of items to which a group is subordinated may have a name of its own, namely, the parent group name.

A group relation may consist of a single repeating or non-repeating parent group entity and a single repeating or non-repeating dependent group entity. On the other hand, a group relation may have multiple dependent groups. In this case, the instances of the different dependent groups can be treated as members of a single set whereby they may be

processed jointly in one context as a single set, or they can be treated as members of several different sets whereby they may be processed independently in another context. This characteristic is not present among compound group attributes.

If the group relation is provided in a given system, it may be an explicitly defined structure type or it may be defined as part of a larger structure type. In either case, the structures possible in the system for group relation composition, the way in which the relations are composed, or the way in which they are used to compose other structures, can vary. Other group relation attributes can include type, identification, and security.

(d) Entry

A particular set of groups in which one and only one designated group is not contained in or subordinate to any other group. The entry corresponds very much to "record," a term not used in this section because of possible confusion with the physical storage structure concept of the same name. Attributes pertinent to the entry can include identification, type, security, and composition.

(e) File

A set of entries that have the same logical organization. The file corresponds to a set of application entities, such as countries, governments, projects or organizations. The entities of a file may be from the same class, e.g., countries, or from different classes, e.g., projects and organizations. Its entries may be explicitly interrelated, or related only by ordering and be otherwise independent of each other. The aggregation of all files which can be accessed by a DMS is the data base. File attributes can include identification, type, security and composition.

(2) Attributes Per Structure Class

Naming the topical sets of attributes for these categories, such as identification, type, composition and security, is a quantitative rather than qualitative process in that these sets are common to the structural classes of most generalized data management systems. The values that the attributes themselves can assume, however, vary from system to system in quality and quantity and should be examined when evaluating one system for a particular capability or when comparing several systems. For instance, attributes of identification which may be of significance to the prospective user, such as provision for synonyms and the number of synonyms permitted, vary among systems.

An explanation of these basic sets of attributes will illustrate how they apply to each structural class and how they can vary in their implementation from system to system. Understanding these factors is also basic to the understanding and evaluation of the data handling functions of a data management system.

(a) Type

Distinguishing different types of items, groups, group relations, entries or files involves determining whether the structure can have different values for the same set of attributes or different usages within the same system. One system, for example, may provide only a single type of file for all processing, while another may provide several files to be used during different functions. Most systems have files differing in usage but not in attributes; e.g., a transaction file, a master file, or a log file, although differing in usage, must have the same set of attributes for identification.

Entries, groups, group relations and items are more likely to have different sets of attributes within the same system. The first three structures are often typed by their attributes of composition, that is, the way in which they are composed or the way in which they can compose other structures. Entries can differ in this respect by the kind of group or group relations that compose them. It is possible for an entry to consist of a single compound group, with hierarchic relationships achieved through group nesting. Another entry can be composed of a set of hierarchic group relations; another of a combination of hierarchic and non-hierarchic group relations. Although all of these types are possible and may vary among systems, within one given system, there is usually only one type of entry provided.

Four types of groups may be identified, the first two provided by all data management systems and the second two provided by many:

- o Non-repeating group - a group for which only one instance of the items/groups that comprise it exists within an entry.
- o Repeating group - a group for which a number of instances of the items/groups that comprise it can exist within an entry.
- o Simple group - a group composed of a single, named collection of items. Simple groups can be repeating or non-repeating.
- o Compound group - a group composed of a collection of a set of items and a set of groups. A reference to a compound group is a reference not only to the set of items, but also to the items of its constituent groups. The groups themselves may be simple or compound and be nested to any depth in a compound group. Within a compound group, hierarchic relationships can be established between two constituent groups, much the same as in the case of the group relation, except that the relationships in a compound group must be hierarchic. Compound groups can be repeating or non-repeating.

Group relations are typed chiefly by the way in which they compose other structures, in this case, the way in which they provide for relations between groups or sets of groups. These relations can be both hierarchic and non-hierarchic. In systems which do not provide the group relation structure, hierarchic relations are provided through the compound group type. There are two kinds of hierarchic group relations:

- o Sequential or linear. Each group, except the first and last in an entry, is related to the group preceding it and the group following it. There is only one group at each level and each group has one and only one subordinate group.

- o Tree. Each group may be related to a number (limited by the system) of groups at any level below it, but to only one group above it in the hierarchy. A parent group, in this case, may have more than one subordinate group but each subordinate group may have only one parent group. In some tree structures, a group relation may also exist between two groups on the same level of subordination. Such groups are called "peers." Because systems differ in the number of dependent groups that may be represented and in the number of times a group may occur, the depth and breadth of the trees can vary greatly, from system to system (see Composition, Section II. 1. a. 2. d).

A group relation can also provide another, more general type of data structure in which the restrictions of a pure hierarchy do not apply. Such a data structure, often called "network," can include not only hierarchical group relations (a dependent group may have only one parent), but also non-hierarchical group relations whereby a dependent group may have a number of parent groups. Thus, any given group in an entry can be related to any other; the fact that a group may participate as the dependent member of more than one parent group allows networks to be built.

Several types of items are provided by most systems to permit a more natural representation of data values. Item types are distinguished primarily by the kind of values they can assume. Numeric item types contain a numeric value and can be used in arithmetic operations. All systems permit at least one numeric item type. Different numeric types in a given system sometimes reflect different storage representation; e.g., a data management system may employ IBM System/360 packed decimal and zoned decimal representations. Although the user in this case can control item representation through a declaration of item type, he must still remain aware of any system restrictions which may exist upon the combining of numeric items of different types. In some systems a user is provided with a single numeric type and the system determines the appropriate representation for each item on the basis of the value initially supplied and provides for any necessary subsequent conversions between representations.

String item types are items whose values are a sequence of characters from a finite alphabet. They are typically used to represent alphabetic or alphanumeric data. All systems provide at least one string item type. Again, there are storage structure considerations involved in the length of strings accommodated by a system; e.g., in an O/S 360 environment 1 to 255 EBCDIC characters may be accommodated.

The item value types themselves can have special attributes of their own, a provision which serves to define very clearly the sets of values assumed by an item and which enhances the data representation capabilities of a system. These attributes can be used to validate values being supplied

for the data base items prior to or during a data manipulation function. Within the given system limitations, length attributes can be assigned by the user to an item value in two forms: a fixed length attribute implies that the values in all instances of the item have the same length; a length range attribute implies that the value length may vary between user-specified limits. If a "picture" attribute can be assigned to an item value (usually defined by means of a string of symbolic characters), not only the length or length range of an item type can be specified, but also such constraining attributes as:

- o Characters permitted at each position in the value.
- o Precision (number of decimal places) of a numeric item.
- o Right- or left-justification if value is smaller than the containing item.

Other item value attributes include ranges and lists of specified values that an item value type can assume and special validation routines to be performed upon the value prior to or during a data manipulation function.

Other item types not properly classified as numeric or string which can exist in a system include data, coordinate (such as latitude and longitude) and date. Other special attributes which can be provided for all or some of the item types in a system include:

- o Units in which values are expressed (feet, pounds, dollars, etc.).
- o Usage of item values (e.g., "computational" or "display").
- o Output editing attributes.
- o Input and output conversion attributes.
- o Value synonym lists (e.g., SINGLE for "S"; MARRIED for "M").

(b) Identification

Attributes of identification are quite similar for each structural category. Almost all categories except for the entry are likely to require some means of identification. Restrictions imposed on this process include length of identifier, use of alphabetic or numeric character (or both) in identifier, provision for embedded blanks in identifier and use of synonyms for identifier.

Attributes of identification usually restricted to the item include:

- o Provision for a user supplied output heading for each item.

- o Unique within group or entry restriction, i.e., whether the same item name can be used for a different item in another group and need be unique only within its containing group.

Special group and group relation identification attributes include the provision for user specified sets of items serving as the group identifier or sequencer. Restrictions placed upon this capability include the number of items in each set and whether ascending and/or descending sequencing can be specified.

(c) Security

Many systems provide security and protection attributes to be specified for a particular structural category at a particular level of data access. User defined access locks, for example, can be provided for controlling access to data contained within a category. In addition, codes which identify the programs making use of the category can also be defined by the user. Security attributes can apply to a specified item, group, group relation, entry or file for the protection of data during a query or update function and at a specified level of frequency (e.g., a value in one item; all item values in one group; all groups at a certain level in an entry; all groups in an entry, etc.).

(d) Composition

The set of attributes entitled "composition" is comprised of regulations governing the composition of a given data structure and the way in which that structure can be used to compose other structures. These regulations can include, for example, limitations upon the number and type of composite structures and upon the interrelationships possible between structures. The only structural category which does not have attributes of composition, because it is itself the smallest element of structure in a system, is the item. Groups, group relations, entries and files have similar sets of composition attributes.

Compound group composition can be limited by such attributes as the number of levels of nesting represented, and the number and/or type of constituent groups. Both simple and compound groups can also be restricted by number and/or type of constituent items. Because the compound group provides a way of establishing a hierarchic relation between two groups, another important attribute of its composition is the manner in which its composite groups are interrelated in the hierarchy. This attribute provided by the group and the group relation allows data to be structured in the manner most suitable to each application. In a compound group, the hierarchy can be a tree structure or a linear structure which is previously described under Type for group relations.

Group relation composition has similar limitations upon the groups that compose it. The kinds of inter-group relations provided by this structure (hierarchical and non-hierarchical), however, unlike the case of the group, are an attribute of Type. Entries can also be limited in composition by attributes such as the number and/or type of group or group relations composing it or by number of hierarchic levels it can contain.

Attributes of file composition include number and/or type of file in the data base, number and/or type of entry composing the file, and the absence or presence of inter-entry relationships within the file. Although the entries of a file tend to be independent of one another in the sense that one entry can be processed without reference to another, they can be explicitly interrelated in a manner known to or controlled by the system. A simple form of explicit relationship is the ordering of the file entries on, for example, the values of the entry sequencer. More general relations can also be made

possible whereby non-hierarchic relations are established between groups in different entries, or between entries themselves when they are composed of a single group. This kind of file is often called a "linked" file. The use of inter-entry relations in a linked file can be further restricted to, for example, relations between non-repeating groups only. In contrast, a file whose entries are unrelated or related only by ordering through an entry sequencer is called an "unlinked" file.

(e) Non-Structural Attributes

Non-structural attributes are attributes which are not necessary elements of the structural scheme (identification, type, composition, etc.), which defines the data category. However, they are closely associated with the category and are available to the user to reference or further define the category during data manipulation. Many systems provide, for example, reference counters and date time of change or insertion indicators for files, entries, groups, group relations, and items.

b. Data Definition

Two basic goals of a generalized data base management system are to store all user data in a data base, and to gain independence of the data from the programs that process it. In order to achieve this, systems provide a data definition capability which is a description, input by the system user, of the names, value classes, constituents, relationships and all other attributes of the various data structures to be established. This description is performed by means of a unified facility provided for the definition of the structure of the data to be stored and processed. Through this facility, data can change without necessarily causing a change in all the programs operating upon it (and vice-versa). Furthermore, centrally stored data can be accessed by different groups of users and still be separately managed, standardized and protected.

One attribute of data definition is the language form used: narrative, keyword, separator, or fixed position (described in Language Attributes of Self Contained Systems, Section II. 2. d). In almost all systems, the data definition for each element is done in the same language, with the syntactical details varying depending upon the particular element being defined. The data definition language usually has the same form as, or is closely related to, those used for the data manipulation functions.

The context of the data definition varies among systems. In most systems, the definition is input to the system separately and processed independently to create directories or tables which are referenced in later processing steps. In others, the definition is an integral part of each program, concerned only with the data for that program and compiled with it.

The structure of the data definition itself can vary. The idea of levels in a hierarchical data structure may be explicitly used in the definition whereby the definition of data at one level would appear before or after the definition of data at a higher level. In this case, unless each line of definition contains its own identification, the lines must be input in sequence.

Another attribute of the data definition function which varies among systems is the capability of the system to revise the existing definition (add new items or change an item value length, for example). The data definition attributes which may be revised will be restricted in one system to the addition or deletion of synonyms for item and group names, while another system not only permits changes to attributes of type, length and identification, but also allows changes in location, e.g., deleting or adding a new node to a hierarchy. Some systems permit the deletion and/or replacement of an entire data element definition. Revision may require the user to submit an entirely new definition or it may merely require only the input of the changes. The user may also be responsible for restructuring the stored data by writing a special procedure if the system does not provide for automatic restructuring.

Revision can involve moving or changing the data values themselves. The data may be transferred from the old file to the new or new data may be placed in the file. An old file, in other cases, can be used as "transaction" input, the values then being copied from the input to the restructured file. In some systems, however, only the file creation and update functions result in data move or change.

Another attribute of data definition is the provision for auxiliary definition. This is, for the most part, a host language oriented function. It involves permitting both an overall, primary data definition and, within the framework of the primary ones, individual data definitions oriented toward particular users and programs. The auxiliary definitions do not replace the original definition although they are usually done in the same form. They provide multiple data structures such as different names for items and groups or multiple lengths or types for a given item value. The relationships that the auxiliary definitions have to the system may vary from system to system. In some, all may have equal status, whereby the system can allow multiple entry definitions. In this case, each definition usually must correspond exactly to the structure of the data as stored, e.g., when a set of items is omitted, a new dummy set having the same total length must be defined in its stead. Some systems, however, allow sub-sets of items to be defined and set up automatic type and length conversion. In contrast to the case of equal-level definitions, other systems permit a primary definition and a number of auxiliary definitions, e.g., the system may provide for a single primary definition applying to the entire data base and independent of particular users or programs. Each user would then be allowed to define an individual data structure within the framework of the primary one.

The auxiliary data definition capability provides several advantages. The auxiliary definition does not necessarily describe the entire data base but only those portions of the data base known to one or more specific programs. Furthermore, it describes them in the form in which they are known to the specific programs. For example, for a user's applications program which defines a working area, an auxiliary data definition can describe the format and characteristics of the data as it appears in the user working area. The data in the program's user work area may then be manipulated using the facilities of the host language. The program is thereby limited to the portion of the data made known to it, a fact which ensures the privacy and integrity

of the rest of the data base from that program. On the other hand, the program itself is protected in that certain changes may be made to the data base without affecting the programs using that data. This is possible because the portions of data described by the auxiliary data definition for the particular program may vary greatly from other portions described by other definitions. Another advantage of this capability is that the user is freed from concern with the entire data base; he need only concentrate on the sections of the data base which are relevant to the program he is writing.

Under the heading, Definition of Components, are enumerated constituent attributes of each data structure class which can be defined by the user. A description of the structure of the data definition statements unique to the definition of the various constituents is not included; such information is best obtained from a users' manual. Certain attributes such as security and identification will nearly always be defined for each data class. The greatest variance among systems is in the methods and requirements for defining group and item value attributes. Most of these attributes, although previously described in Logical Structure, will be reviewed in the context of data definition.

Attributes of items for which systems can require definition include:

Identification - the definition of the name of the item and any synonyms to be used for the name. Some systems also require an item number which can be used in place of the name in the definition and elsewhere in the system.

Security - to provide against unauthorized access at the item level, a user must usually supply certain codes before being allowed to use the item. The requirement may differ for query and update. Sometimes the user may specify the name of a security checking procedure at the item level to be used in connection with item references.

Item value attributes - the attributes of numeric and string item values outlined in Logical Structure, if provided by the system, also require definition. They may be specified either at the same time as the item name or in a separate section of the data definition. If not an intrinsic part of the item definition, they are considered as part of the validation which must be applied to the data values used in the creation and update function.

Possible attributes include a statement of the range of values permitted, a minimum or maximum value, or a specification of, by listing them, the actual values which the item may assume. The user may also be able to set up synonyms for certain values. He may be able to control the right- or left-justification of the item in an area longer than necessary to contain it.

Several attributes, particularly item value type, length, positioning and editing directions, may be combined into a picture statement which contains a string of one-character codes each defining the characters which may occur in the corresponding position of a value of that item. For example, the picture statement "$$999V99" defines the item value length (five decimal digits) by the 9's, the position of the decimal point by the V, and a floating dollar sign for editing by the $$.

The length of the item may be unnecessary to define, if it is fixed in the system. A statement of the starting character position of the item value in the containing group or entry may be required in addition to length. Subitems in this case are string item value types which may be permitted the possible definition requirements being starting and ending positions, subitem name, and length.

The definition of groups involves the organization of the definitions of its constituent items and/or groups as well as attributes of the whole. The relationship of the group to its constituents if it is a simple group may be accomplished explicitly by referencing the group in the definition of the item or vice-versa. It can also be accomplished implicitly through the ordering of definitions so that, for example, a group is defined to consist of all items whose definitions immediately follow or precede the statement defining the attributes of the group itself. If the group is compound, its definition must include the definitions of the contained groups. Systems vary in this case, some requiring interwoven item and group definitions and others requiring a separate definition of item and group attributes.

Pertinent to the facility for hierarchical organization is the definition of group or item level numbers. Often each group in a hierarchy is assigned a level number which indicates how far down in the hierarchy it is, thus allowing the system to reconstruct the hierarchy from a set of sequentially presented definitions. Another attribute of definition ordering is whether the "top-down" or "bottom-up" approach is used, that is, whether the top levels or the lowest levels of the hierarchy are defined first.

Other attributes of the group which can be defined include:

- Count items. Items in a parent group, one for each dependent group containing a count of the number of instances of the dependent group.
- Special keyword. A word used in the definition statement referring exclusively to a repeating or non-repeating group.

Attributes of group relation definition which vary among systems (for those which provide a definite group relation structure) concern the explicit or implicit definition of the relation. In systems allowing only a superior-to/subordinate-to relation, the definition of the relation is implied by the definition of the hierarchical entry structure. Systems providing an explicit definition vary in that the definition can either be separate from the group definition or part of it.

Entry definition includes an identifier in almost all systems. However, very few separate attributes aside from entry identification is really the definitions of its constituents. For example, the definition of relationships between groups of an entry is accomplished in the group definition.

Similarly, a file definition consists mainly of a union of its constituent entry, group and item definitions. To be noted is the sequence of these constituent definitions which can vary among systems. Separate attributes of file definition usually include file security and identification.

c. Storage Structure

Storage structure concerns the data as it is actually stored on the available physical media and the various methods provided by which stored data is accessed. This element of data structure is distinguished from logical structure which in contrast involves the user's conception of his data in terms of its logical organization. A single data structure can be stored in different ways, resulting in different storage structures. Many systems select from several different storage techniques depending on how the data is to be used. Because a given storage structure may be implemented in different ways on different storage devices, the nature of the storage structure is essentially conditioned by the system's host environment dependencies, that is the characteristics of the storage devices and of the operating system under which the system must function. Such pertinent characteristics as hardware environment features and features of the hardware and operating system environments are described in Host Environment Interrelationships (Section II.4). In Storage Structure are identified such attributes of storage structure as user control over storage structure, storage organization techniques, data address accessing and operating system supplied facilities for storage structure control and access.

There are eight major file structures presently in use. These data structures are as follows:

- o Sequential Access
- o Indexed Sequential Access
- o Tree
- o List
- o Chain
- o Multilist
- o Inverted
- o Network

Each of these structures generally is considered a "pure" form in the industry literature. No attempt is made here to discuss the myriad variations and hybrids which in practice comprise the majority of structures actually used in data storage and management. Each of these has a certain logic behind its formulation, but all can be reduced to some combination of the structures just listed.

The following segment provides a detailed breakdown of DMS attributes relative to data structure and definition.

I. Data Structure and Definition

A. Logical Structure

1. Items:

a. System term for item
b. Item types provided:

(1) Numeric item types:

(a) System term (packed decimal, fixed binary, right justified EBCDIC, etc.)
(b) Storage representation
(c) Length
(d) Numeric sub-items permitted

(2) String item types:

(a) System term (alphanumeric, left-justified EBCDIC, variable length, etc.)
(b) Storage representation
(c) Length
(d) String sub-items permitted

(3) Other item types:

(a) Date
(b) Coordinates
(c) Boolean

c. Item value type attributes:

(1) Numeric:

(a) Fixed length
(b) Length range
(c) Picture:

(1 Positioning
(2 Precision
(3 Right/left justification

(d) Value range
(e) Value list
(f) Validation routine

(2) String:

(a) Fixed length
(b) Length range

(c) Picture:

(1 Positioning
(2 Right/left justification

(d) Value range
(e) Value list
(f) Validation routine

d. Item identification:

(1) Length
(2) Mode
(3) Embedded blanks
(4) Synonyms
(5) Unique within:

(a) Group
(b) Entry

(6) Output heading

e. Security:

(1) Access locks controlling access during:

(a) Query function
(b) Update function

(2) Program identification codes for use of item during:

(a) Query function
(b) Update function

f. Other item attributes:

(1) Units of expression (feet, pounds, dollars, etc.)
(2) Usage of item values
(3) Output editing attributes
(4) Input and output conversion attributes
(5) Value synonym lists

g. Non-structural attributes:

(1) Existence indicators
(2) Date and time of change to item value
(3) Identification of transaction or program supplying current item value
(4) Null item values
(5) Multiple item values

2. Groups:

a. System term for group
b. Group types:

(1) Simple repeating
(2) Simple non-repeating
(3) Compound repeating
(4) Compound non-repeating

c. Group identification:

(1) Effective length
(2) Mode
(3) Embedded blanks
(4) Synonyms
(5) Use of group identifiers or sequencers:

(a) Required or optional
(b) Number of items used
(c) Ascending and/or descending sequence

d. Security:

(1) Access locks controlling access during:

(a) Query function
(b) Update function

(2) Program identification codes for use of group during:

(a) Query function
(b) Update function

e. Group composition:

(1) Number and type of constituent items
(2) Number and type of constituent items and groups in compound group
(3) Depth of nesting in compound group
(4) Hierarchical structure composed:

(a) Linear
(b) Tree
(c) Number of levels of subordination
(d) Number of dependent groups per parent group
(e) Number of peer groups at same level of subordination

f. Non-structural attributes:

(1) Date and time of group insertion
(2) Count of references made to group

3. Group relations:

a. System term for group relation
b. Group relation type:

(1) Non-hierarchic
(2) Hierarchic:

(a) Linear
(b) Tree

c. Group relation identification:

(1) Length
(2) Mode
(3) Embedded blanks
(4) Synonyms

d. Security:

(1) Access locks controlling access to parent or dependent groups during:

(a) Query function
(b) Update function

(2) Program identification codes for use of parent of dependent groups during:

(a) Query function
(b) Update function

e. Group relation composition:

(1) Number of levels of subordination of composite groups
(2) Number of dependent groups per parent group
(3) Number of peer groups at same level of subordination
(4) Placement criteria for insertion of new constituent group occurrence

f. Non-structural attributes:

(1) Count of constituent groups or repetitions of a constituent group

4. Entries:

a. System term for entry
b. Entry type:

(1) Single type entry only
(2) Multiple entry types provided

c. Entry identification:

(1) Length
(2) Mode
(3) Embedded blanks permitted
(4) Synonyms permitted
(5) Item sets serving as identifier or sequencer

d. Security:

(1) Access locks for controlling access to entry instances during:

(a) Query function
(b) Update function

(2) Codes which identify programs using the entry during:

(a) Query function
(b) Update function

e. Entry composition:

(1) Limitation on number and type of group and group relationships comprising the entry
(2) Limitation on the number of hierarchic levels in the entry
(3) Limitation in number of relations in which a dependent group may participate in the entry

f. Non-structural attributes:

(1) Date and time entry was placed in the file
(2) Count of references to entry

5. Files:

a. System term for file
b. File types:

(1) Single file type only
(2) Multiple file types provided:

(a) Differentiated by function
(b) Differentiated by sets of attributes

c. File identification:

(1) Length
(2) Mode
(3) Embedded blanks
(4) Synonyms:

(a) Maximum number of synonyms
(b) File can be accessed by synonym (not restricted to output)

d. Security:

(1) Access locks for controlling access to file during:

(a) Update function
(b) Query function

e. File composition:

(1) Number of files in data base
(2) Number and type of entries in file
(3) Limitations on inter-entry relations

f. Non-structural attributes:

(1) File instance numbers
(2) Date-time stamp
(3) Entry counts
(4) Control totals over file entries

B. Data Definition

1. Data Definition Language

a. Form used (narrative, keyword, fixed position, separator)
b. Form same as language used by other DMS functions
c. Same language is used to define all data elements

2. Context of Data Definition

a. Function is an integral part of each program processing the data and is compiled with it
b. Function is integrated with file creation function
c. Function is input to system as a separate function
d. Lines of data definition can be input in any order

3. Data Definition Revision

a. Re-entrance of entire definition required
b. Input of changes only required
c. Data values are moved or changed during revision
d. Restructuring of stored data must be provided by user
e. System provides automatic restructuring of stored data

f. Existing definition may be deleted for which attributes of:

(1) Item
(2) Group
(3) Group relation
(4) Entry
(5) File

g. Existing definition may be expanded or modified for which of the above (I-B-3-f-1 through 5) data elements
h. Entire definitions may be deleted or replaced for which of the above (I-B-3-f-1 through 5) data elements

4. Definition of Components

a. Item definition:

(1) Identification:

(a) Item name
(b) Item number
(c) Synonym

(2) Security:

(a) Requirements for user access during:

(1 Update function
(2 Query function

b. Item value definition:

(1) Item value type
(2) Item value length:

(a) Specified in digits, bytes, characters
(b) Fixed in system

(3) Item value placement (left- or right-justification):

(a) Starting character position of item value
(b) Definition of value range:

(1 Minimum/maximum value
(2 Item value list

(c) Attributes defined in picture statement
(d) Editing directions for input/output or for validation:

(1 Defined in picture statement
(2 Specified by separate routines or tables

(3 Capabilities include:

-a- Automatic truncation
-b- Automatic padding
-c- Additional information added to data
-d- Encoding of values
-e- Left-justification of field
-f- Zero suppression
-g- Truncation
-h- Algebraic sign
-i- Punctuation
-j- Percent
-k- Engineering notation
-l- Dollar sign
-m- Specific characters
-n- Decoding of values
-o- Data conversions
-p- Standard measurement conversions
-q- Scaling
-r- Rounding
-s- Item size modification
-t- Right-justification of field

(4) Item value definition is specified at same time as item definition
(5) Item value definition is in separate section of the data definition

c. Subitem definition:

(1) Identification
(2) Length
(3) Starting position

d. Group definition:

(1) Item and group attributes defined separately for:

(a) Repeating group
(b) Non-repeating group

(2) Relationship of constituent item/group definitions to parent group:

(a) All have same level or group number
(b) Definition ordering (group is defined to consist of all items whose definitions immediately follow or precede the group definition statement)
(c) Top-down or bottom-up approach

(3) Identification
(4) Security:

(a) Requirements for user access during:

(1 Update function
(2 Query function

(5) Group sequence item:

(a) Number of items in sequencer set
(b) Ascending or descending sequence may be specified

(6) Explicit group level number:

(a) Group and items or groups only

(7) Count item
(8) Special keyword for group:

(a) Repeating group
(b) Non-repeating group

e. Group relation definition:

(1) Identification
(2) Security:

(a) Requirements for user access during:

(1 Update function
(2 Query function

(3) Definition is implicit (it is implied by the definition of the hierarchical entry structure)
(4) Definition is explicit (ordering is stated in definition statement)
(5) Definition is separate from group definition
(6) Definition is part of group definition

f. Entry definition:

(1) Identification
(2) Security:

(a) Requirements for user access during:

(1 Update function
(2 Query function

(3) Organization of constituent groups:

(a) Specified in group or group relation definition
(b) Specified separately in entry definition

(4) Multiple logical data structures can be defined within one entry

g. File definition:

(1) Identification
(2) Security:

(a) Requirements for user access during:

(1 Update function
(2 Query function

(3) Sequence of constituent definitions

C. Storage Structure

1. Control

a. Host environment dependencies:

(1) File storage device is automatically assigned if no special request is made
(2) DMS uses operating system supplied access methods
(3) DMS uses operating system supplied space and resource management (allocation of buffers, provision of additional storage in the event of overflow)
(4) Operating system provides for storage device interchangeability

b. User control of storage structure:

(1) Devices available (see HOST ENVIRONMENT INTERRELATIONSHIPS):

(a) Sequential devices
(b) Direct access devices

(2) Sequential or direct access can be chosen:

(a) Operating system supplied access methods can be chosen
(b) DMS augments O/S supplied access methods

(3) User control over space allocation for file on storage medium:

(a) Length
(b) Variable or fixed
(c) Blocking factor
(d) Maximum and average size
(e) Number of entries per record

(f) Entry length
(g) Paging technique (storage subdivided into specified number of characters)

(4) User control over index arrangement
(5) User supplied random accessing formula is necessary

2. Storage Structure Representation

a. Logical sequential (each group instance is stored according to its subordinate relationship starting with the master group)
b. Sequential (elements of data follow one after the other but do not conform to the logical organization of the file):

(1) All values of an item are stored at any vacant place in the file, saving the address of that location for further use in retrieval)

3. Organization Techniques

a. DMS augments O/S supplied access methods to provide better indexing for hierarchical files
b. Techniques for establishing logical sequential organization include:

(1) Table index (an ordered reference list to the contents of a file)
(2) Chaining:

(a) Groups are chained together using top-down method (chained to next group at same level)
(b) Groups are chained together using top-down method (dependent groups follow superior groups; the last group in one string is chained to the first group of the next higher level string)
(c) Chaining is combined with table indexing
(d) Pointers are comprised of:

(1 Address of a physical record
(2 Address of a group
(3 Item value

(e) Multiple pointers are used
(f) Storage allocation for pointers

4. Storage Access Methods

a. Indexing (the association of a data value with the address of an element containing that value):

(1) Performed through operating system access methods:

(a) O/S provides overflow for index entries not fitting into memory

(2) DMS provided indexing features:

(a) Index contains one value-address pair for every occurrence of a given value
(b) Index references only ranges of values
(c) Inverted file used -- file indexed by every value of every item

b. Randomizing or hash coding -- the transformation of item values in the group or entry into an address:

(1) User supplied randomizing formula necessary or may be specified to replace O/S supplied formula
(2) Formula can handle synonyms (records whose control field happens to randomize to the same address as other records):

(a) Synonyms are placed in overflow area provided

(3) Formula is affected by file size
(4) Formula is affected by volatility or stability of data base

2. Data Manipulation Functions

a. File Creation

The file creation function provides the creation of the initial instance of a file, making it known to the DMS which will perform other functions on it. This process can involve entering a data definition for a file which already exists in machine readable form so that it becomes acceptable to the DMS. It can also involve the more complicated task of converting an existing file into a form acceptable to the system. Such a conversion can be performed by a self-contained function provided by the system through a special use of the update function or through a facility provided specifically for the purpose of file creation.

Several steps can be defined as constituting the creation action process in most systems. They are as follows:

- o Data definition of the master file and its constituent structures (see Data Definition, Section II. 1. b).

- o Storage structure definition of the master and input source file. This step can be provided automatically or be specified in part by the user. A choice of access methods to be used on the input data file is usually provided to the user. (See Storage Structure, Section II. 1. c.)

- o Data definition of the input source file. This step uses facilities provided for data definition of the master file or is accomplished by means of a special definition capability to accommodate transactions which are like those used in the update function. (See File Update, Section II. 2. b, for a description of transaction processing.)

- o Allocation of media space for the file. This step accomplishes the reservation of storage space for files and can be achieved by means of a statement of data definition which provides parameters to the operating system space allocation facilities. (See Host Environment Interrelationships, Section II. 4.)

- o Provision of data on the input file. The input file is the source from which the data comes for the population of the master file. Attributes pertaining to this step include the kind of files accepted as input and the host environment dependencies involved in supplying the input files. Specific examples include:

  - o Inter-system capabilities, the acceptability of input data files generated on other computers or under different operating systems.

  - o Intra-system capabilities, the acceptability of input files produced by other system processors within the operating system under which the DMS operates.

- Acceptance of files constructed by the use of the interrogation function from existing system files.

- Acceptance of data input as a stream of transactions without the user being aware of an intermediate file if, for example, a conversational mode of input is supplied to the system.

- The types of local and remote hardware devices that can supply the input data.

- Multi-file input capabilities.

The input file may have to be specially prepared in a system required format, or the system may provide facilities defining the data and storage structure of data already in machine readable form. In the latter situation, the system may perform certain data transformations within entry instances or it may process data in its existing form. In some systems, input file provision can be aided by the interrogation function facilities to determine the size of the file to be created and to establish need for editing, restructuring or any other necessary change in the data when this capability is not available during file population.

- Population of the file. This final step is achieved by defining an existing file so that it becomes acceptable to the system or mapping data from the input file to the master file. The method employed for this process may be either the update function or a function specialized for the purpose of creation. This task is usually accompanied by the following procedures:

  - Data validation. At data definition time, the user can specify the types of validation to be performed on each item; he can also specify data values against which data entering the system can later be compared. During file population, the system can then compare each entering data value against the predefined validation criteria and accept only those values satisfying the criteria. The user can usually modify or override the criteria either permanently or temporarily during operations other than data definition.

  - Data transformation of input entries or items to the format of the master file, for example, transforming the order of items, groups or entries from the order which they have in the input file to some other order in the file being created. Other transformations include changing the values of item instances. New items can be derived from old items through arithmetic functions, or they can be encoded or decoded by table substitution or by special programs. These transformations may be specified at the time of file definition, by procedural statements supplied to the population step, or at the time of input data provision.

- o Error detection and reporting features, for example, testing for duplicate entries or entries invalidated by inter-entry tests in the input file; logging or dumping of master file contents to provide recovery in case of hardware failure during the process; collection of counts or values of source data or master file data for inter-entry validation or control totals.

An important attribute of the file creation function, in addition to those pertinent to the particular steps, is the capability to monitor the creation cycle so that both the errors encountered in each step and the statistics concerning the size and resources of files used can be adequately reflected. Errors reported usually include faults detected by the validation criteria, diagnostic messages on violations of language syntax, and data unacceptable to the input/output facilities of the operating system. Corrections of invalid data may be provided in on-line or batch mode. Facilities of the interrogation function may also be available to the user to monitor certain creation activities. These attributes are similar to those which characterize File Update monitoring. A more complete coverage of monitoring, error reporting, and restart and recovery procedures is given in DMS System Control (Section II. 3).

b. File Update

Updating a file is the process of using update data to change values in all entries or selected entries, groups or items stored in the file. It does not include changing the logical data structure, data validation criteria or security procedures. (Alterations of this type are usually made by revising or redefining the data definition of the data base.) The File Update function is supplied by the Data Definition function with a description of the section of the data in the data base that is to be updated and by the File Creation function with the data currently stored in the section of the data base to be updated.

File Update is common to all data management systems in at least one mode. Often a system will provide more than one update mode depending upon differences in user control, input media used, language used, or functions provided. For example, a system can have one mode whereby update processing can be specified for any of the data structures in the file and another mode whereby changes can only be made to a certain data structure, such as file items. The attributes of the File Update function described below can vary not only from system to system but also from mode to mode within a single system.

The major elements of File Update are the update data (transactions), the description of the update data to be applied to the data file (transaction definition), and the processing rules or algorithms that apply the update data to the data file (transaction program). These elements have several sets of attributes which are likely to vary from system to system. One set of attributes common to all three are their sources or the way in which they can enter the system. The input medium available for transactions must be considered as well as the time at which the transaction definitions and programs must or can be entered. Batch processing may either require the transaction definitions and programs to be supplied at the beginning of the batch or

permit them to be included in the transaction. Such requirements also prevail when transactions are entered from a remote terminal. In many systems the transactions and their definitions and programs may be prestored, a capability requiring a separate run prior to the update run whereby they are stored in a form and on a medium that makes them accessible to the system. Another attribute closely associated with sources of input and modes of processing is the ordering of transactions. Update processing can be speeded up, particularly in the case of systems that use sequential files, by processing transactions in the same order in which the affected entries are stored in the data base. Transaction ordering may be a user responsibility. Some systems accept and process the transactions in any order or, if necessary, sort them into file sequence before processing.

Attributes of transaction definition include format and placement. The transaction may be defined in one of several available formats, such as narrative or fixed tabular, which can also be employed for the transaction program applied to the same transaction. The format(s) in some systems are prescribed while in others any format may be acceptable as long as the definition has been previously stored and is accessible by the system. The placement of the definition is another variable. A language capability may be provided that permits transactions to be submitted to the system in a self-describing form. In this case the transaction and its definition are intermingled. On the other hand, especially in systems which use the same medium for both the transaction and its definition, a complete separation of the two is required, the definition preceding the transaction. Such techniques may vary within the system according to mode or according to the particular transaction data element (item, group, entry) that is named.

Format and placement are also attributes of the transaction program. In some systems the transaction program may be intermingled with the transaction definition in the same format. In others the definition must have been previously entered into the system. Data mapping, whereby the transaction group and item names are equated to corresponding master file group and item names to which they correspond, is another possible attribute of the transaction program. In some systems, this is achieved by using file group and item names in the transaction definition for corresponding data elements rather than using transaction programming statements. Use of the transaction definition for this purpose is usually not possible, however, when data files created for other purposes are used as transactions.

The group of attributes belonging solely to the transaction program element concerns the access and manipulation of data. These attributes are extensive and must be broken down into several sets. The first of these sets includes possible characteristics of the amount of user control over data access. The retrieval of transactions and file entries with matching identifiers may be done automatically with no user specification. Systems providing this type of automatic transaction and entry access usually write out the updated entries automatically. In other systems complete control may be provided to the user by requiring him to specify each step of the data access process. This may include instructions for reading individually each group contained in a transaction from the input medium. After the update process

has been completed, the user may also be required to locate, read, and write out entries by using instructions that operate on data elements below the entry level.

Data selection is the identification in the transaction program by the user of the part of the file that is to be changed by a transaction. One method of data selection is through a statement of the logical relations that must be satisfied before a transaction is applied to the file. Such statements are usually in the form of conditional expressions which can be in the same language used by the interrogation function. (Attributes of conditional expressions are described under Language Attributes of Self-Contained Systems, Section II. 2. d.) Another method is for the system to match the transaction to the entry through matching group or entry identifiers without further specification of selection criteria.

Data base changes involve both arithmetic and non-arithmetic changes. Arithmetic change capabilities are those provided for the addition, subtraction, multiplication, division, etc., of item values, constants, and literals. Pertinent characteristics include the use of literals in arithmetic operations, the use of item values in the arithmetic computation of the values of other items and the specific arithmetic operators available. Non-arithmetic changes involve data value modification and the deletion and addition of items, groups or entries. Consideration should also be given to the capability of the system to detect and report errors encountered during this process.

The data base changes possible and the requirements for achieving them vary not only from system-to-system, but also from one data element to another within a single system. Updating on the entry level implies a transaction which takes into account all the data in a file entry. Changes to an entry can include insertion, whereby a new entry is added to the data file or whereby new groups from the transaction are placed in the file entry where there are no matching groups. Entry insertion can be accomplished through data manipulation statements or through an automatic insertion where there is no matching entry identifier in the file. Another change can be a transaction that identifies an entry to be deleted from the file. The entry identifier in some systems may also be updated. Errors detected by the system can include, for example, checks for a duplicate entry already in the file if insertion of a new entry is to take place and a list of rejected transactions upon user request.

Group level changes imply a transaction which takes into account all items or subordinate groups within a group. Groups also can be updated through deletion, insertion and identifier change using the same or different statements as for the entry. Insertion can involve the replacement of matching groups or the addition of new groups which do not have a matching identifier in the file. In some systems group insertion involves setting up an empty group, the insertion of data being accomplished by item level changes. An important attribute of group deletion, in addition to the statements necessary for this change, is the way in which groups subordinate to the one being deleted are handled.

Item level update involves to a large extent value changes performed by arithmetic functions or by non-arithmetic data manipulation statements, such as changes to individual characters or to substrings of characters within the item. Another type of value change can take place when an item value changes to or from an established system null value. In some systems this is referred to as the deletion or insertion of an item. In most cases items can be physically inserted or deleted only when groups or entries are stored in variable length physical records. The system can inform the user when no match is found for an item in the file when a value change is to take place. Information supplied by the system to the user during this process can include notification of a no match condition between a transaction item and a corresponding file item or the placement of a value in an item which is the same as the previous value of that item.

Transaction validation, editing and transformation can be accomplished either through the transaction program or through the transaction definition facility. Validation may include simple checks on item values to ensure that they are within the limits supplied by the user in the transaction definition (see Data Definition, Section II. 1. b). Conditional expressions used for the interrogation function may also be available for the user to specify, for example, logical relations that must exist within a transaction before it is applied to a file or logical relations that must exist between the transaction and the file data before the transaction can be applied to the file. Editing is usually provided through data definition facilities. Transformation may be also supplied through encoding and decoding subtables specified at transaction definition time or through computational algorithms applied during the transaction program to the data.

An important attribute of the update function, applicable to all its elements, is the automatic maintenance of system integrity through adequate system monitoring and error detection facilities. This provision can include audit trail features, backup files, run histories, and lists of rejected transactions. Errors detected by the system may be under conditions predefined by the user as in the case of data validation attributes specified at file definition time. Other system detected errors may include language syntax errors, logic errors, processing errors germane to a particular update mode, and comparison time errors (e.g., a transaction that is received for an entry not in the file or an attempt to insert an entry with an identifier that is the same as one already in the file). The system may permit the user to specify a course of action to be taken if a certain condition should occur, for example, terminate the run, print error message, ignore transaction, etc., if there are transaction data errors, language syntax errors, no matching identifier, etc.

Operating system features provided for the maintenance of system integrity, such as the prevention of interference from the execution of other programs using the same files, are covered under Host Environment Interrelationships (Section II. 4). The maintenance of file storage is often provided exclusively by the operating system although in some cases it is accomplished by the data management system which can provide checks for available storage space, terminate the run if space is not available and then provide regenerated storage space through a utility or subprogram. A more

complete list of DMS monitoring and error detection attributes is described in DMS System Control (Section II. 3).

c. Interrogation

The interrogation of a data file takes place in three basic steps: the selection of data based upon the satisfication of user specified conditions, the extraction of the selected data, possibly for optional intermediate processing such as sorting or summarization; and the formatting of the results into reports or into a machine readable form for further processing. Many systems refer to these steps in three separate sets: data selection, data extract, and report formatting. The characteristics of other features which can be provided by the interrogation function, such as sorting and generating files for further processing, are also included in this section.

The explicit capabilities of the language used in the interrogation function for data selection and transformation, provided by the logical and arithmetic operators, operands, connectors and conditional expression features, are described in Language Attributes of Self-Contained System (Section II. 2. d). In self-contained systems (systems containing a built-in processing algorithm to obtain user specified information so that he does not have to program or control the steps the system must use to process his requirements), although these language attributes are more closely identified with the interrogation function, they are usually available to the file update and creation data selection processes. The difference in language operations among the functions pertains to the action statements used after the data has been selected. These attributes must be treated separately.

The action statements used by the update function control the change in value content to be made at some level in the data file. Those used by interrogation cause data to be extracted and placed either into report form or into machine readable form possibly after intermediate processing. These actions are essentially non-procedural, for instance, such actions as the control of data transfer between levels of memory are not controlled by the user but are built into the self-contained system in the form of an implicit processing algorithm. However, some self-contained systems provide explicit statements to open and close files, to initiate data transfer, or to store data in temporary areas in high-speed memory. In a host-language system, the user usually programs the series of conditions and actions which define the process by which the information required is to be generated. In this chapter, the attributes described are mainly characteristic of a self-contained system, although in some host-language oriented systems, similar facilities may be provided for non-procedural interrogation or non-procedural report generation.

Attributes of the data selection process described in this chapter are inferred from the capabilities of the language used for data selection. They characterize the ability of the system to locate a specified element of data at a specified level of logical organization. This ability is the direct result of the degree of precision attained by the conditional expression in establishing search criteria. A conditional expression for instance, may be composed in such a way that any item from any hierarchical level in a file

may be located or that once an item in a particular group has been found, all groups subordinate to that group qualified by the search item can also be located for further processing. Other attributes of data selection include the system facility used for this process (like the update function, the interrogation function may be accomplished through several processing modes within the system) and the actual language statements used for data selection.

Data extract is often not clearly delineated in the analysis of a system because selection and extract are not mutually exclusive. Often extract is regarded as essentially the copying of selected data before it is formatted into a readable report. In other cases, the term "extract" is treated synonymously with "select" and both head the same set of attributes. However, the two concepts can be more clearly separated for the purpose of analysis by postulating two different sets of attributes. Selection may be said to involve the depth of the search achieved within the limitations imposed by the rules and language tools available for composing a conditional expression. Extraction involves considering which quantities at each level (item, group, entry, file) can be searched for and placed in the report or output file. Extraction also decides how many files or reports associated with the same file can be output for the same conditional expression. The allowance for multiple outputs may also involve deciding the direction of data to different output media.

Output presentation is the formatting of the results of the user queries into readable reports or into a machine readable form for further processing. This section describes the two modes of output usually present in a DMS and enumerates the possible report formatting capabilities of each which can be provided by the DMS. On-line, off-line capabilities are also mentioned in that this capability should be available on multiple and diverse output devices; a more complete description is given in Host Environment Interrelationships (Section II. 4).

The two modes which can be specified within a system are the standard system supplied formats which are an integral part of the system and can be provided to the user either automatically or upon request, and the user composed report mode (also called the report generation mode) which provides a means for the user to compose his own output formats and can be tailored to provide the exact output formats and contents desired The report generation mode can, for instance allow the user to cause the output to conform to preprinted or to extremely wide output forms. Both modes can be available under recurring scheduling (production at a specified frequency) or demand scheduling. Validation and editing capabilities are also available to both modes.

Specific attributes of the standard output mode include:

- o A standard set of report types, i.e., the way in which all the data is presented (tabular, row).

- o The automatic calculation and adjustment of the format parameter to the output device; e.g., column-width, the presentation of outputs in balanced columns which are automatically adjusted to the

number of characters in a line of output to correspond to the maximum number of characters per line that can be printed by the output device.

- Parameter specification - system can select and print headings provided by the user such as report title, column heading, date, time, page numbers, etc.

- Special functions. The system can provide functions specified by the user such as sums and counts of retrieved item names and values if specified by the user; counts and tallies of given item occurrences; statistical operations on given values, and editing and decoding operations.

The report generation or user composed report mode relies on specific report creation capabilities provided by the DMS to the user for the data to be output. These capabilities can include:

- The use of literal values and item values for headers.

- Pagination control, whereby the DMS provides for user specified parameters for controlling and numbering individual pages of an output report. Examples include starting page specification; the user is allowed to specify the starting page number and the increment to be used in determining the page number of succeeding pages, page break; the user is allowed to specify certain methods to stop printout on a given page and begin a new page independent of specified line counts.

- Data reduction features, whereby DMS allows user specified sums, counts and statistical operations to be performed on specific item names or values.

- Page headers and trailers.

- Outputs, whereby DMS allows user to cause multiple copies of a report or to cause output to conform to certain output forms.

- Special outputs, whereby DMS provides to user upon request special outputs such as job summaries either on high-speed printer or at a terminal.

Other capabilities which can be performed by the interrogation function enable the user to produce or create data values and retrieval specifications at one point in a sequence of operations to be used at a later time, e.g., during output presentation. This capability includes the production of temporary files which can be used as an input file in another statement or to produce additional copies of previous outputs or reports; the creation of prestored retrieval specification; the modification of those specifications; and the parameterized execution of those specifications.

d. Language Attributes of Self-Contained Systems

This section identifies the possible capabilities of the data handling languages provided by the self-contained system to its functions concerned with selecting and transforming data. A system may have multiple languages or sub-languages which vary in form. These forms can be divided into four types:

- Narrative - an English-like language form which, though having syntactic descriptions, resembles English sentences.

- Keyword - a language form consisting of a sequence of attribute-value pairs. Languages for systems that operate in an interactive mode are usually of this type.

- Separator - a language form like the keyword form whereby there is a keyword followed by a sequence of attribute-values separated by some special character.

- Fixed position - language form in which each element of the definition appears in a fixed position (e.g., punched card column) on an input medium. Often a preprinted form or questionnaire is provided to simplify the user's task.

The major portion of this section describes the procedure oriented capabilities that the language(s) may possess. Included are arithmetic and logical capabilities of the language as performed by conditional and arithmetic expressions. The composite elements of such expressions are identified:

- Operands

  A data entity upon which an operation is performed. It may be simple, i.e., a literal value, an item or the results of some computation, or it may be a combination of simple operands. When the values of two items are compared, the two operands may be different items or the same items. They may be selected from the same logical entry or from two different logical entries.

- Operators

  The devices used for expressing the relationship between operands. This includes conventional and special comparators, functional operators, arithmetic operators, data reduction operators, Boolean operators and geographic search operators by which item values in geometric areas can be located based on geographical coordinates.

- Complexity of Expressions

  The level of complexity of expressions indicates the number and variety of combinations of expressions. This capability involves the provision for compound logical expressions, levels of nesting

within expressions, quantitative limitations on formation of compound expressions from simple expressions and mixed mode expressions.

The following segment provides a detailed breakdown of DMS attributes relative to data manipulation functions.

II. Data Manipulation Functions

A. File Creation

1. Input file definition

a. System specified format required
b. User specified format permitted
c. Same facilities provided for definition of master file
d. Same facilities provided by update function for defining transactions
e. User can specify access method for input file
f. Data and storage structure definition performed in single task

2. Allocation of media space

a. Operating system facilities provide for:

(1) Space allocation for input file
(2) Space allocation for master file
(3) Interchangeability of device type
(4) Overflow areas
(5) Diagnostics

b. Utility programs provided
c. User specifications required:

(1) Device type
(2) Blocking method
(3) Space reservation for:

(a) File size
(b) Overflow area
(c) Working area

3. Provision of input data file

a. Acceptability of input data files generated on other computers or under different operating system:

(1) Foreign files accepted
(2) Physical media:

(a) Magnetic tape
(b) Disc
(c) Other

b. Acceptability of files produced by other system processors within the operating system under which the DMS operates:

(1) COBOL
(2) FORTRAN
(3) PL/1
(4) ALGOL
(5) Other

c. Acceptability of input data constructed from existing system files by the use of the interrogation function
d. Acceptability of input data as a stream of transactions
e. Hardware environment:

(1) DMS can accept input data from the following local sequential devices:

(a) Card reader
(b) Paper tape
(c) Magnetic tape

(2) DMS can accept input data from local keyboard devices
(3) DMS can accept input data from the following remote sequential devices:

(a) Card reader
(b) Paper tape
(c) Magnetic tape

(4) DMS can accept input data from the following local direct access devices:

(a) Drum
(b) Disc
(c) Data cell

(5) DMS can accept and incorporate format descriptions for input data from:

(a) Card reader
(b) Paper tape
(c) Magnetic tape
(d) Keyboard device
(e) Drum
(f) Disc
(g) Data cell

(6) DMS can accept and incorporate file descriptions from:

(a) Card reader
(b) Paper tape
(c) Magnetic tape
(d) Keyboard device
(e) Drum
(f) Disc

(g) Data cell

(7) DMS can accept and incorporate file descriptions from a remote:

(a) Card reader
(b) Paper tape
(c) Magnetic tape
(d) Keyboard device

f. Multi-file input capabilities
g. Format requirements:

(1) Direct correspondence between the sequence of the input data and the logical organization of the file to be generated
(2) Several different formats for the input data during file creation are permitted
(3) If data is in machine readable form, system can review it to determine transformation, restructuring or editing requirements
(4) File may be prepared through use of interrogation function

4. File population

a. Accomplished through update function
b. Accomplished through separate creation function
c. Validation facilities provided:

(1) Item value validation criteria (see Data Definition)
(2) Sequence check of entries
(3) Size check of entries

d. Data transformation:

(1) Transformation of item values
(2) Encoding, decoding of item values
(3) Reordering of items, group, entries

e. Monitoring and error detection facilities provided during file population

5. Creation function monitoring

a. System detected errors
b. Operating system detected errors
c. User specified conditions for error detection
d. On-line corrections permitted
e. Batched corrections permitted
f. System statistics reports
g. Restart and recovery procedures provided

h. Run history of recognized errors provided

B. File Update

1. More than one update mode exists within the system characterized by differing:

   a. Functions
   b. Languages
   c. Input media
   d. Amount of user control

2. Sources

   a. Input medium for transaction (see FILE CREATION, Provision of Input Data File)
   b. Batch processing may use:

      (1) Prestored information:

         (a) Transactions
         (b) Transaction definitions
         (c) Transaction programs
         (d) File update procedures may be prestored:

            (1 In source form
            (2 In object form (system library)

         (e) Prestored update procedures may be modified either permanently or temporarily:

            (1 Stored library form (the procedure as stored on the library is modified and the change is permanent)
            (2 Temporary modification can be made at run time and is effective only for the current run

         (f) Parameterized procedures can be prestored and the parameters entered at execution time

      (2) At the beginning of the transaction stream:

         (a) Transaction definitions
         (b) Transaction programs

      (3) In the transactions:

         (a) Transaction definitions
         (b) Transaction programs

   c. Remote terminal processing may use:

(1) (Same as II-B-2-b above)

d. Transaction ordering:

(1) User responsibility required by the system:

(a) For all processing modes

(2) Transactions can be processed in any order
(3) System performs presort

3. Transaction definition

a. Facilities described under Data Definition available to transaction definition
b. Format and placement:

(1) Format(s) used (fixed, tabular, narrative, etc.)
(2) System required format
(3) Separation of transaction and transaction definition required
(4) Format and placement requirements differ according to:

(a) Update mode used
(b) Transaction data element defined

(5) Data mapping is a function of transaction definition
(6) Data validation features provided
(7) Data editing and transformation features provided

4. Transaction program

a. Facilities described under Language Attributes of Self-Contained Systems available to the transaction program
b. Format and placement:

(1) Formats used
(2) System required format
(3) Separation of transaction program and transaction definition
(4) Format and placement requirements differ according to update mode used

c. Data mapping is a function of the transaction program
d. Data validation features provided
e. Data editing and transformation features provided

5. Data access and manipulation

a. Data access:

(1) Automatic capabilities (no user specification):

(a) Automatic transaction and entry access by matching identifier
(b) Automatic file reading
(c) Automatic file writing
(d) Automatic capabilities differ according to update mode used

(2) User control:

(a) File reading:

(1 User control requirements differ according to update mode used

(b) File writing:

(1 User control requirements differ according to update mode used

b. Data selection:

(1) Achieved through statement of logical relations to be satisfied before the transaction is applied to the file:

(a) Conditional expressions used:

(1 Capabilities described for conditional expressions listed under Language Attributes of Self-Contained Systems available to data selection process

(2) Achieved through matching of transaction and entry identifiers with no further specification of data selection criteria

c. Data change operations:

(1) Data manipulation facilities described under Language Attributes of Self-Contained Systems available to data change process
(2) Arithmetic changes:

(a) Use of literals; e.g., add (100, 000) to population where 100, 000 is a literal and population is a name
(b) Computation of a new data value (the capability to compute a new data value from other data values)
(c) Arithmetic operators used include: *, =, +, /, **
(d) Arithmetic changes can be performed on value contained in all occurrences of a specific item

(e) Specific errors detected by the system during arithmetic change attempt

(f) Supplemental information reported by the system during arithmetic change attempt

(3) Non-arithmetic changes:

(a) Insert (the capability to insert physical values, sets of values, or entries to a file that has previously defined their logical counterparts) of:

(1 Item:

-a- Update mode(s) used
-b- Data change operators used
-c- Specific errors reported by system during item level insert
-d- Supplemental information reported by system during item level insert
-e- User specified errors or information to be reported
-f- User specified course of action under predefined conditions

(2 Group:

-a- Update mode(s) used
-b- Data change operators used
-c- Specific errors reported by system during group level insert
-d- Supplemental information reported by system during group level insert
-e- User specified errors or information to be reported
-f- User specified courses of action

(3 Entry:

-a- Update mode(s) used
-b- Data change operators used
-c- Specific errors reported by system during entry level insert
-d- Supplemental information reported by system during entry level insert
-e- User specified errors or information to be reported
-f- User specified courses of action

(b) Delete (the capability to delete physical values, sets of values or entries to a file that has previously defined their logical counterparts) of:

(1 (Same as those listed under II-B-5-c-3-a)

(c) Replace (the capability to replace physical values, sets of values or entries to a file that has previously defined their logical counterparts) of:

(1 (Same as those listed under II-B-5-c-3-a)

(d) Identifier changes for:

(1 (Same as those listed under II-B-5-c-3-a)

6. File update monitoring and error detection facilities:

a. System detected errors
b. Operating system detected errors
c. User specified conditions for error detection
d. On-line corrections permitted
e. Batched corrections permitted
f Run history of recognized errors provided
g. List of rejected transactions provided
h. System statistics reported
i. Audit trail features provided
j. Backup file features provided
k. Maintenance of file storage during update:

(1) Performed by operating system
(2) Performed by the DMS

C. Interrogation

1. Query Language:

a. Sufficient reperitoire of comparators and connectors is provided including:

(1) Operators
(2) Logical connectors

b. Expressions entail:

(1) Complex expressions
(2) Levels of nesting
(3) Combined expressions
(4) Mixed arithmetic/boolean expressions

c. Sufficient mathematical functions are provided
d. Procedural language actions:

(1) Open and close of files
(2) Data transfer between levels of memory
(3) Specification of arithmetic functions

(4) Definition of temporary storage in high-speed memory
(5) Statement labeling
(6) Statements transferring execution control
(7) Statement looping

2. Data Selection

a. System facility used
b. Data selection statements available
c. Depth of search:

(1) Locate specified data element from any level in logical organization of the data base:

(a) Retrieve from a specified instance of a repeating group
(b) Retrieve from all repetitions of a repeating group

(2) Distinguish between the absence or presence of data values at any level of data

d. Capability to terminate a search based on a specified number of group instances qualified
e. Capability to use the results of a single selection as input values to a subsequent selection
f. Capability to locate groups of data that satisfy search criteria and allow the use of the following rules to define related data groups that qualify for retrieval:

(1) Locate only those instances of a group whose item values satisfy the search criteria
(2) Locate a superior group only if all instances of a subordinate group satisfy the search criteria
(3) Locate a superior group if at least one instance of a subordinate group satisfies the search criteria
(4) Locate all subordinate groups of a qualified group
(5) Locate a superior group based on a subordinate group that is more than one level removed
(6) Locate any group related to a group that satisfies the search criteria
(7) Locate those instances of a group whose items do not satisfy the primary search criteria but satisfy alternate or secondary search criteria

3. Data Extract

a. System facility used
b. Data extract statements available
c. Data elements extracted (data value)
d. Attributes of data element other than data value which can be extracted:

(1) Name
(2) Type
(3) Length
(4) Date stamp
(5) Existence status
(6) Other descriptions stored at data definition time

e. Extract data elements used as search criteria in conditional expression
f. Extract text through:

(1) Association with formatted data
(2) Content based on search criteria

g. Multiple routing of user-selected outputs
h. Optional routing or output to devices other than user's terminal
i. Capability to extract particular sets of data values for additional processing (e.g., sorting) based on the evaluation of the conditional expression

4. Storage and Recalling of Interrogation Statements

a. Statements may be stored, recalled and modified for one use only
b. Statements may be stored, recalled, modified and restored under a new name for multiple uses, without destroying the original statement
c. Statements may be stored and recalled with the user supplying parameters to complete the query at execution time
d. Statements created for later execution can be composed:

(1) At a console
(2) In a batch mode

e. User can request execution of a pre-stored query directly from:

(1) Remote consoles
(2) Local consoles

f. Statements composed on-line are:

(1) Executed directly
(2) Batched for execution
(3) Both

5. Temporary File Creation

a. Establish a temporary file for additional processing which is a subset of and has the same logical structure as the file from which it was produced for additional processing:

(1) Temporary file can be used as an input file in another function or output presentation

b. Establish a temporary file which is not the same logical or physical structure as the system file:

(1) Temporary file can be used to produce additional copies of previous outputs or reports

6. Report Formatting

a. Standard mode - output presentation in selectable standard formats:

(1) Available under demand scheduling
(2) Available under recurring scheduling (outputs produced at a specified frequency, e.g., daily, 3 times daily, weekly, etc.)

b. Report generation mode - ability for user to compose desired output formats:

(1) Available under recurring scheduling
(2) Available under demand scheduling

c. User can specify:

(1) Recurring scheduling for standard mode output
(2) Recurring scheduling for report generation mode output
(3) Demand scheduling for standard mode output
(4) Demand scheduling for report generation mode output

d. All validation and editing capabilities are available in:

(1) Standard output mode
(2) Report generation mode

e. Output presentation capability includes as a library function the ability to compose:

(1) Cover pages
(2) Title pages
(3) Distribution lists
(4) Special handling instructions

f. On-line/off-line capabilities provided:

(1) On-line output media:

(a) Typewriter
(b) Display
(c) Teletype

(2) Off-line output media:

   (a) Tape
   (b) Disc
   (c) Printer
   (d) Card punch

(3) Audio:

   (a) Spelled voice
   (b) Spoken voice

(4) Automatic conversion to account for the specific characteristics of any given device is provided:

   (a) Standard method for representing characters that do not exist on the device is included

7. Standard Output Mode Features

a. Standard formats are:

   (1) Parameter-driven
   (2) Invoked by name

b. Standard report set includes the following report types:

   (1) Tabular-column presentation of data
   (2) Row-oriented (a simple row-by-row list of data values with indentation to indicate their position in the hierarchy)
   (3) Functional formats:

      (a) Presentation of counts
      (b) Summations

   (4) File format (a presentation of the names and relationships of the levels of the file)

c. Automatic formatting capabilities include:

   (1) Column/row width
   (2) Column/row spacing
   (3) Line width
   (4) Number of lines per page
   (5) Above parameters can be automatically calculated and adjusted to the output device capacity
   (6) User can specify parameters for the above formats as an override option

d. User can specify the following parameters when initiating standard formats:

(1) Report title(s)
(2) Security classification:

    (a) User authorization
    (b) Highest classification of output data

(3) Date or as-of-time
(4) Page numbers
(5) Column headings:

    (a) Item name (item name appears at head of column of its values)
    (b) Query specified (user specified title replacing or added to the item name)
    (c) Data description title (heading other than item name specified in the data description)

(6) Trailer information (information provided at bottom of each output page, e.g., page number and security classification
(7) Table of contents based on report titles

e. Special functions which can be selected by user:

(1) Counts of item occurrences
(2) Counts of unique item values
(3) Counts of particular item values
(4) Counts based on conditional queries
(5) Sums of particular item values
(6) Percent total
(7) Different reports (system can generate different report formats from one retrieval statement)
(8) Multiple copies (system can provide more than one original copy of the same report)
(9) Statistical functions of item values:

    (a) Mean
    (b) Median
    (c) Mode
    (d) Standard deviation

(10) Editing functions
(11) Decoding functions
(12) Capability to override decoding transformations (obtain stored data value)
(13) Subtotals
(14) Subcounts

8. Report Generation Mode Features

a. Headers - ‛he capability to employ for titles, row headers and column headers, the following:

(1) Literal values
(2) Item names

b. Positioning - the capability to specify for titles, column headings, row titles and items, the following:

(1) Vertical position
(2) Horizontal position
(3) Data value position
(4) Right-justification
(5) Left-justification

c. Pagination specification - the capability for the user to specify the following parameters:

(1) Starting page
(2) Page increment number
(3) Page number
(4) Last page
(5) Location of page number
(6) Page break on:

(a) Sort key
(b) Value of a sum
(c) Number of lines on a page
(d) Particular count value
(e) Output form

(7) Termination of output based on:

(a) Number of pages
(b) Number of lines
(c) Item values

(8) In the absence of these specifications, several standard options are provided

d. Data reduction features include:

(1) Maximum item value
(2) Minimum item value
(3) Statistical functions of specified item values:

(a) Mean
(b) Median
(c) Mode
(d) Standard deviation

(4) Counts of item occurrences
(5) Counts of all unique item values
(6) Counts of particular item values
(7) Sums of all occurrences of an item

(8) Percent total

e. Page headers and trailers include:

(1) Security classification
(2) Page number
(3) Date
(4) Time
(5) Report titles
(6) Table of contents
(7) Column headings:

(a) Item name/number
(b) Title specified in query
(c) Title for column heading in data description

f. Outputs

(1) Use of preprinted forms
(2) Spread sheet output (e.g., output is two physical pages wide)
(3) Different report formats from one retrieval
(4) Multiple copies of a report from one retrieval

g. Special outputs available upon request:

(1) Control file reports:

(a) List of stored procedures
(b) Specific logical file organization
(c) Items in a specified group
(d) Data definitions of:

(1 Coding
(2 Transformation
(3 Validation criteria
(4 Security

(2) Job summaries:

(a) Rejected data
(b) Validated transaction data
(c) Invoked procedures
(d) Error conditions
(e) Statistics (e.g., number of entries processed, volume of transaction data, etc.)

D. Language Attributes of Self-Contained Systems

1. Language Form:

a. Narrative

b. Keyword
c. Separator
d. Fixed position
e. User defined changes to the language:

   (1) Synonyms for verbs
   (2) Syntax changes
   (3) Semantic changes

2. Language Operands

   a. Simple operands:

      (1) Item
      (2) String value
      (3) Numeric value
      (4) Variable
      (5) Functions
      (6) Square root
      (7) Natural logarithm
      (8) Results of computation
      (9) Trigonometric:

         (a) Sine
         (b) Cosine
         (c) Tangent
         (d) Arcsine, arccosine, arctangent

   b. Compound operands:

      (1) Any combination of simple operands
      (2) Different group instance; same item
      (3) Different entry; same item

   c. Transformation of standard operands:

      (1) System can transform the following operands to present them consistently with their file counterparts:

         (a) Date
         (b) Time
         (c) Temperature
         (d) Location
         (e) Pressure
         (f) Distance
         (g) Volume

3. Language Operators

   a. Basic relational operators:

      (1) Equal

(2) Not equal
(3) Greater than
(4) Less than
(5) Greater than or equal
(6) Less than or equal

b. Special operators:

(1) Greater than but not blank
(2) Less than but not blank
(3) Character pattern
(4) Absence
(5) Between
(6) Mask match
(7) Maximum/minimum
(8) Empty
(9) Increase/decrease

c. Functional operators:

(1) Increase
(2) Decrease
(3) Product
(4) Sum

d. Arithmetic operators:

(1) =
(2) +
(3) -
(4) /
(5) *
(6) **

e. Mode of computation permitted:

(1) Floating decimal
(2) Decimal
(3) Integer

f. Reduction operators:

(1) Count
(2) Total
(3) Average
(4) Partials (subtotals, subcounts)

g. Logical connectors:

(1) AND
(2) Exclusive OR
(3) Inclusive OR

(4) NOT

h. Geographic search:

(1) Circle
(2) Convex polygon
(3) Concave polygon
(4) Straight-line intersections
(5) Combinations of the above

4. Statistical Operations

a. Calculate:

(1) Arithmetic mean
(2) Mode
(3) Median
(4) Standard deviation of a field

b. Count:

(1) Number of unique values of a field
(2) Number of occurrences of a particular field value

5. Conditional Expressions

a. Types of conditional expressions permitted:

(1) Logical
(2) Arithmetic
(3) Boolean
(4) Combination of the above

b. Natural evaluation of expressions (e.g., left to right scan of the following symbols: (), *, -, +, and exponentiation)

c. Complexity of conditional expressions:

(1) Capability provided to mix arithmetic and boolean expressions
(2) Operand may be mixed mode
(3) Number of conditional expressions that can be combined directly
(4) Number of levels of nesting using parentheses
(5) Minimum of 40 operators and operands per expression
(6) Compound conditions on same item within expression
(7) Precedence rules for logical connectors within expression
(8) Logically connected conditions on several items but with the same reference quantity within expression

d. Conditions may be specified for:

(1) Items
(2) Groups
(3) Entries

e. Expression can be given an identifier

3. DMS System Control

DMS system control involves the ability of the system through its communications and housekeeping capabilities to be both controlled and used efficiently and to be evaluated for the planning of modifications directed toward improving its performance. Pertinent capabilities include monitoring, error recording, restart and recovery procedures, and security. Scheduling capabilities in multiprocessing, multiprogramming and multi-user environments are outlined in Host Environment Interrelationships (Section II. 4).

The monitoring capability provides an optional recording of DMS activity. Recording control pertains to the use of recording categories which define a particular collection of events to be recorded. Every event to be monitored in the DMS is assigned to a category. The highest system recording category records all events defined. Lower categories record a particular subset of specified events. A system usually provides for varying the recording frequency based on the system recording category in operation or upon such criteria as specific time of day and specific sampling period.

Information gathered covers two categories of system use, data base access and program module use, and can cover both general statistics and specific facts. The general statistical information provided by monitoring are tallies of events such as total items retrieved and tallies of the number of consoles in use and the number of disc seeks issued; total times for events such as total time for processing a specific job or total time required to search the file; and standard job accounting or job history which may supplement standard operating system accounting.

Specific information can be provided by two modes of monitoring: demand and background. Demand monitoring involves the surveillance of a condition or conditions specified by the system manager and provides for both standard and specialized queries of system activity at the discretion of the system manager.

Standard queries report on such system activity as what devices and modules are currently in use, what type of file is being accessed, what users are currently active and what amount of working storage is available. Specific queries single out particulars of system activity such as data accessed by a specific user within a given time period or the time of the last update performed on a specific file.

Background monitoring occurs on a continuing basis in order to maintain system regulations. It informs the system manager of the existence of abnormal system use such as attempts to access the data base illegally or the reasons for abnormal termination of runs.

The DMS should also have an error recording capability which provides for data base accountability, the assurance of processing integrity, the full identification of errors and the correction of error conditions. All user initiated communications including data input is subject to system error detection and validation, including checking for format, syntax and semantics. The DMS should be able to record errors by the operating system such as

hardware generated errors. The provision for diagnostics is also important to assist the system user in checking out new procedures and developmental files. This provision is actually a test mode of operation which facilitates debugging operations as well as protecting the system against possible simple and catastrophic failures resulting from test failures.

The DMS should have capabilities necessary for the recovery from interrupts and from programmer, operator and hardware errors. To provide for the event of catastrophic failures, the system should also be able to maintain a backup data file.

Another pertinent attribute of system control is the provision of security features to protect the information contained in the data base from illegal access. These features can include restriction of access through invoking read/write protection at various levels of data, automatic destruction of data on any storage device in the event of imminent compromise, the automatic clearing of any area of core containing classified information immediately after the last use of that information, and the assignment at file definition time of access locks or security codes to files, entries, groups and items.

Of further interest to the user are the time security clearance takes place, the levels of data at which security restrictions can be defined by the user, the extent of security restriction application (data access and/or data modification), and the security clearance procedure to be satisfied by the user himself before he can execute any data handling statements.

The following segment provides a detailed breakdown of DMS attributes associated with DMS system control.

III. DMS System Control

A. Monitoring

1. Recording Control

a. Capability for user control (user can specify types of events to be recorded)
b. Number of recording categories provided
c. Capability for system to vary the recording frequency based on:

(1) Time of day
(2) Time interval
(3) Sampling rate
(4) Sampling period
(5) System recording category in operation

2. General Information Recorded

a. Capability for generating and recording the following types of statistics:

(1) System tallies:

(a) Item retrieved - count of the number of items retrieved
(b) System module executed - a list of which system modules have been executed and how often
(c) Device usage - tallies of the number of consoles in use and the number of disc seeks issued
(d) Input transactions - a tally of the number of transactions applied to a file and of those transactions which were rejected

(2) System event times:

(a) Job execution - total time for processing a specific job
(b) System module execution - total processing time for each system module used
(c) Data access - total time required to search the file

(3) Job history:

(a) Capability to provide a history of all DMS jobs on demand
(b) History information includes:

(1 Programmer definition
(2 Job time:

-a- Time on
-b- Time off
-c- Duration

(3 Job termination:

-a- Successful
-b- Unsuccessful
-c- Abnormal termination reason

(4 I/O time by I/O event
(5 CPU time by task
(6 DMS modules executed by execution sequence
(7 O/S modules executed by execution sequence
(8 I/O device usage by access sequence
(9 Data accessed or modified
(10 Record of operator intervention

(c) Capability for off-line storage of historical data

3. Specific Information Recorded

a. Data base access:

(1) Capability to record for any level of data, the following:

(a) Who initiated the access
(b) What program modules were employed, including, for each employment:

(1 Time in
(2 Time out

(c) What level of data was accessed

(2) Capability to define new data to be collected
(3) File backup capability for recovery purposes or statistical analysis

b. Program modules:

(1) Capability to obtain for each program module used:

(a) User identification
(b) Initiation time
(c) Termination time
(d) Other information as needed

4. Monitoring Levels

a. Provision for demand monitoring
b. Provision for background monitoring on a continuing basis

(1) Demand monitoring:

(a) Standard queries:

(1 What queries are currently active
(2 What type of file is being accessed
(3 What is the classification of the file being accessed
(4 Amount of working storage available
(5 Which devices are currently in use

(b) Specialized queries:

(1 Capability to specify:

-a- Which events are to be recorded
-b- Conditional recording of events
-c- Dynamic overriding of recording

(2 Specialized queries can include:

-a- Itemization of all users who have accessed a specific file within a given time period
-b- Time of last update performed on a specific file
-c- Person who performed the update
-d- Data accessed by a specific user within a given time period
-e- A list of all files currently in the system and their classification, size and resident storage device
-f- A list of all users authorized access at a given security level and all files included in this classification
-g- Itemization by name of all procedures created by a specified user, providing them for selective viewing on a CRT terminal

(2) Background monitoring

(a) Capability for system manager to monitor the system for instances of abnormal system use:

(1 Attempts to access system operation capabilities
(2 Attempts to access the data base illegally
(3 Abnormal termination of runs

B. Error Recording

1. System Detected Errors

   a. Item values - includes invalid characters, incorrect number of characters, invalid values, etc.
   b. Transaction format - includes invalid field lengths, incorrect sequence of values, invalid transaction codes, etc.
   c. Procedural statements - involves syntax or punctuation of procedural statements

2. Operating System Detected Errors

   a. Equipment malfunctions - hardware generated errors
   b. Task or job specification errors - errors in requesting the execution of a task or job, e.g., incorrectly naming the program to be executed, incorrect device specification, etc.

3. Diagnostics Provided

   a. Snapshot dumps of work areas
   b. System program dumps
   c. Breakpoint capability:

      (1) Conditional termination
      (2) Conditional execution

   d. Trace of the procedural program
   e. Indication when a file or procedure is not acceptable for incorporation into the system
   f. Trace of item values at any level of aggregation which is being accessed

4. User Specifications for Dumps and Traces

   a. User sp[illegible]d output media for dump or trace
   b. User can cause dumps to occur or traces to begin and end a[illegible] points within the processing which are:

      (1) Predefined
      (2) Conditionally defined (when procedure is compiled)
      (3) Defined at execution time for batch mode
      (4) Dynamically defined on-line

   c. User specified amount of output from a dump or trace
   d. User specified type of information that is made available for each element of the dump or trace
   e. User specified memory areas to be dumped:

      (1) Absolute memory locations
      (2) Symbolic names of:

         (a) Jobs
         (b) Procedures

(3) Parts of jobs or procedures
(4) Several non-contiguous areas of memory by a single dump command

f. User specified data base areas to be dumped:

(1) File names
(2) Group names
(3) Item names
(4) Several portions of a data base in a single dump statement, e.g., two different files

g. User specified traces of:

(1) Procedures and subroutines
(2) Times:

(a) Start
(b) End
(c) Duration

(3) Input arguments:

(a) Names
(b) Values

(4) Items updated:

(a) Item identification
(b) Old item value
(c) New item value
(d) Time of update

(5) System resources used:

(a) Type of resource, e.g., CPU component, tape unit
(b) Specific identification of resource

h. User specified limitation of trace of:

(1) Procedures
(2) Data
(3) Resources

i. User specified limitation of trace to:

(1) Specified occurrence intervals
(2) Specified logical conditions
(3) Specified maximum trace lengths
(4) Specified points for trace to begin or end in terms of:

(a) Occurrence intervals, e.g., begin trace after 50th subroutine has been executed
(b) Times, e.g., cease trace after 10 minutes

C. Restart and Recovery

1. Detection and recovery provided from errors caused by:

   a. Programmer
   b. Operator
   c. Hardware
   d. System

2. Error messages contain recovery instructions automatically or upon request
3. File backup capability for restart or recovery from data base damage:

   a. Generation of log file of transactions entering system from terminals
   b. Generation of log file of messages sent by system to terminal
   c. Storage needs for backup, e.g., extra tape drives

4. Processing interrupt:

   a. Processing may be abandoned at any time by the user
   b. Processing of a user program may be suspended at any time by:

      (1) The system operator
      (2) The user

   c. Recovery from job suspension provided through:

      (1) Full save provisions
      (2) Full restart provisions

   d. Processing can be terminated by user at other than normal termination points
   e. Portions of a program normally processed can be skipped on a temporary (one time only) basis

D. Security Features

1. Security restrictions can be applied at file definition time at:

   a. File level
   b. Entry level
   c. Group level
   d. Item level

2. Security restriction applies to:

   a. Data access
   b. Data modification

3. Clearance takes place:

   a. At the opening of the data base
   b. At time of issuing data handling language statements

4. Clearance levels provided for any level of data:

   a. Unclassified
   b. Confidential
   c. Secret
   d. Top secret

5. Number of access categories within each security level
6. Specification of a higher classification for an entire report than the classification of the report components
7. Capability to have classification level names adjusted by the user
8. Capability to have classification level names abbreviated
9. Provision of an automatic destruction capability for:

   a. All on-line storage devices
   b. Remote disc packs
   c. Remote tapes

10. Necessitation of physical intervention for destruction of:

    a. Internal storage
    b. Non-internal storage, e.g., tape, disc

11. Provision of read protection for:

    a. File
    b. Entry
    c. Group
    d. Item

12. Provision of write protection for a:

    a. File
    b. Entry
    c. Group
    d. Item

4. Host Environment Interrelationships

Identified in this section are the possible aspects of the data management system's dependency upon or interrelationship with its host environment, that is, procedural language programming facilities, operating system facilities, and hardware configuration.

a. Host Language Attributes

This section should pertain only to the study of systems which are built upon procedural languages such as COBOL or PL/1 as opposed to systems providing only the self-contained functions described earlier (data definition, query, output presentation, creation and update) which do not require conventional procedural programming. The programming facilities embedded in the procedural language, which is the host language, represent capabilities distinctly different from those of the self-contained functions. Attributes of these facilities included here have been obtained exclusively from one source: Feature Analysis of Generalized Data Base Management Systems, a report compiled by the CODASYL Systems Committee in May 1971 (see Bibliography). A thorough, well-organized explanation of the attributes of host language programming facilities, which are merely outlined here, can be found in Section 7 of this report entitled, "Programming Facilities."

b. Hardware Environment

Some systems are designed to operate only on one particular configuration, while others are designed to be machine independent. The attributes identified within the area of hardware configuration include the following:

- Minimum hardware configuration. When considering the minimum hardware configuration which is used to support the system, the processor involved, the minimum memory provided for both on-line and batch versions, and the required hardware options must be included. Such a consideration might find, for instance, that the minimum configuration merely allows space for the operating system of the particular machine plus the particular DMS without provision for buffers.

- Data base storage media. Attributes of storage media include required storage for the operating system alone, additional requirements for the data management system, and other secondary storage supported for data base storage, e.g., extra tape drives to allow backup for restart and recovery from data base damage.

- Terminal equipment. Various types of terminal equipment are used with on-line systems. Considerations should involve the terminal types available, the number of terminal types that can be active at one time, and the location of the terminal.

c. Operating System

Data management systems rely on operating systems to varying degrees. Pertinent attributes of this relationship to be considered include operating environment, scheduling, modes of processing available and software interface.

- o Operating environment. An important attribute of the operating environment is the method of scheduling a group of programs or run units. Any single program may be in one of these states:
  - o Running (being serviced by a CPU).
  - o Blocked (unable to run because it is waiting for some other action to be completed before it can be rescheduled).
  - o Waiting (ready to run, but in a queue until the CPU or other resources are available to it; it will enter into execution when it has the highest priority of all other waiting jobs).

Depending upon the design of the operating system, there may be one or several programs co-resident in the machine. They may be run concurrently or separately. They may also run for a fixed amount of time and be terminated for rescheduling, or be rescheduled only when blocked. Three categories can be used to describe such possible features:

- o Uniprogramming system is one in which a program is initiated and the scheduler component of the O/S does not start another program until the first one is completed.
- o Multiprogramming system allows all three states of a program to exist (running, blocked and waiting).
- o Multiprocessing system is one having more than one processor. Each CPU of such a system can adopt the attributes of either a uniprogramming or multiprogramming system.

d. Scheduling

Scheduling involves the ability of the operating system either to provide scheduling facilities in a multiprogramming, multi-user environment or to supplement existing DMS facilities. The handling of the problem of concurrency, when two or more users may be running programs which endeavor to perform one or more functions (create, update, query) concurrently on the same or different portions of the data base, must be considered. Can the operating system, for example, handle the simultaneous creation of two files? What methods are used so that concurrent functions can occur when only one copy of the DMS exists? Can concurrent operations be performed on different data files?

- o Software facility interfaces. Outlined here are attributes of the possible operating system functions available to the DMS which

affect the way in which the DMS performs or is implemented. These include input/output facilities, remote processing control and other facilities such as sort/merge and the compiler used in host language systems and for own-code routines in self-contained systems.

- Mode of system use. Modes of system use include interactive and batch. These affect the way that data and procedures can be entered and information retrieved. The interactive mode is of two types which may be supplied by a system for some or all system functions. The first type is the conversational mode, whereby a user engages in a dialogue with the system, usually on a question/answer basis in which he responds to system provided questions or options in order to execute his request. It is found in systems that lead the user through the steps of a terminal session, or upon request tell the user what alternatives he has at a specific point in a terminal session. The second type of interactive mode is the prestore. It occurs when the terminal user is allowed to examine data and prestored procedures and to specify execution procedures, data, and/or parameters that differ from those that were prestored. Unlike the case of the conversational mode, the system does not actively assist the user. He must know beforehand what to do and how to do it.

IV. Host Environment Interrelationships

A. Host Language Attributes

1. Levels of Interface Provided:

   a. Narrative or free form of writing down manipulation language statements
   b. Fixed tabular format (less scanning is required)
   c. Call processing module (go directly to processing module, no scanning, parsing, validation or interpretation of data manipulation language statement is required)
   d. Machine oriented interface (statements reference data at the detailed physical I/O level)

2. Programming Modes:

   a. Input (the transferral of data from the data base to the program)
   b. Output (the transferral of data from program to data base, used in file creation or add modification)
   c. Update (encompasses input and output):

      (1) System keeps track of generations of a file

   d. Access mode restrictions (i.e., update can only be performed on random file under COBOL)

3. Access Methods

   a. Sequential
   b. Random
   c. User specified
   d. No mode distinction made by the user

4. Method of Invoking Programming Facilities

   a. Programming facilities can be invoked from any host language
   b. Method of interface is different from one host language to another
   c. Method of invocation:

      (1) Call to control module
      (2) Verb set

   d. Explicit exit required from host language

5. Language Form

   a. Narrative
   b. Fixed position

c. Separator
d. Keyword

6. Addressable Data Structures

   a. File:

      (1) Referencing statement

   b. Entry:

      (1) Referencing statement

   c. Group:

      (1) Referencing statement

   d. Item:

      (1) Referencing statement

7. Program-System Communication

   a. Currency (keeping track of current position in the data base):

      (1) Single pointer
      (2) Multiple pointers
      (3) User must reset all currency pointers in an update
      (4) Pointers are maintained in terms of an internal identifier for each group
      (5) Pointers reference:

         (a) Entries
         (b) Current group (most recent group processed)
         (c) Parent group of current group

   b. Error handling
   c. Selection criteria:

      (1) Selection of data is accomplished by the association of an identifier with the data manipulation language statement
      (2) Selection of data is accomplished by the association of a conditional expression with the data manipulation language statement:

         (a) Conditional expression capabilities for selection

      (3) Form and content of selection criteria:

         (a) Logical and relational operators
         (b) Comparison of item values to:

(1 Constants
(2 Ranges
(3 Other item values

(c) Existence conditions

(4) Selection criteria appear in the data manipulation language statement
(5) Selection is achieved through communication items initialized by the user with values of identifier items

8. Security

a. Security clearance takes place:

(1) At time of opening part of data base for processing
(2) At time of issuing data manipulation statement

b. Program is linked to only part of the data base in a binding process; the privacy of the rest of the data base is automatically ensured
c. Security restriction is applied to:

(1) Data modification
(2) Data access
(3) Both

d. Security is based on:

(1) Authority level
(2) Need-to-know level
(3) Both

e. Security is defined at the following levels:

(1) File
(2) Entry
(3) Group
(4) Item

9. Data Manipulation Language Statements

a. Control statements (no data movement is accomplished)
b. Open statement (begin processing of data file):

(1) Mandatory for referencing data
(2) Identification of part of data base to be processed is required
(3) Identification of processing mode is required
(4) Different restrictions apply to opening of sequential and direct access devices

c. Close statement (finish processing data file):

   (1) Open and close statements are required in pairs
   (2) Different restrictions apply to opening of sequential and direct access devices

d. Conditional statements (modify sequence in which host language statements are executed)

e. Data retrieval statements (data from data base is moved to user working area with no change in data base):

   (1) Locate and access statements:

      (a) Separate statements required for locate and access
      (b) Several retrieval options are available based on:

         (1 Use of selection criteria
         (2 Data level accessed
         (3 Random or sequential access

   (2) Simple access statements (used to make data available in user working area after location is determined, no selection criteria is used):

      (a) Statements unnecessary (system already provides general purpose locate and access statement)

   (3) Hold or reprocess statements (retain data in user working area for future processing or to lock out access by another run before present run is finished)
   (4) Currency reset statements (reset currency pointers)

f. Data modification statements:

   (1) Add:

      (a) Add data to end of file
      (b) Insert data into a file
      (c) Populate a null file
      (d) Different statements used to perform different add capabilities

   (2) Change (change item values within instance of entries and groups which previously existed within the data base)
   (3) Delete:

      (a) Group only
      (b) Group and all dependent groups
      (c) Group and optional dependent groups

(d) Different statements used to perform different delete capabilities

(4) Reorder statements (reorder the sequence of entries in a file or group within a logical string, e.g., sort)

g. Special purpose statements (handling data in a communications environment or in primary storage):

(1) Table handling capability
(2) Communications (transfer groups of items, called messages or transactions between user working area and queues often associated with external terminal devices):

(a) Incoming transactions may be used to interrogate or update the data base
(b) Transfer of transactions is accomplished using the same statement provided for data base manipulation

B. Hardware Environment

1. Minimum Hardware Configuration

a. Processors
b. Minimum memory:

(1) Batch
(2) On-line

c. Required hardware options, e.g., decimal, arithmetic, real time clock, drum storage, etc.

2. Data base storage media

a. Required storage devices:

(1) Operating system only:

(a) Tape
(b) Direct access devices

) Addition for minimum data base system:

(a) Random access devices only
(b) Any device supported by the particular operating system
(c) Tape

3. Terminal equipment

a. Traffic volume:

(1) Maximum number of on-line consoles or terminals that can be connected to the system
(2) Maximum number of consoles that may be active at a given time
(3) Maximum number of on-line users who may have jobs being processed

b. Machine interface
c. System start-up procedure:

(1) Manual (data phoned to remote site, verbal to computer operator)
(2) Automatic (interrupt compiler from on-line console)

d. System sign-off procedure:

(1) Manual
(2) Automatic (sign-off signal to system)
(3) Console input message to remote computer operator

e. Equipment:

(1) CRT:

(a) Dark room required
(b) Output presentation:

(1 En masse
(2 Line-by-line

(c) Cursor:

(1 Destructive
(2 Non-destructive
(3 Cursor can be positioned anywhere on the screen
(4 Any display (contents of CRT screen) can be printed by on-line user:

-a- Typed command (software):

-1- Remote printer
-2- Available printer

-b- Special purpose key (hardware):

-1- Remote printer
-2- Available printer

-c- Size of display:

-1- Number of lines
-2- Characters per line

-d- Available character set

(2) Teletype:

(a) Characters per second
(b) Typing errors corrected by:

(1 Backspace
(2 Erase character
(3 Delete line/command/query

(c) Noise level:

(1 Negligible
(2 Otherwise

(3) Keyboard:

(a) System reserved keyboard characters not available to user
(b) Number of special command keys

4. Recovery procedure for type or format error:

a. Correct a character
b. Correct a line
c. Entire procedure must be reentered

C. Operating System

1. Operating Environment

a. Uniprogramming system
b. Multiprogramming system:

(1) Number of co-resident jobs
(2) Number of simultaneous jobs
(3) Dismissal of a program when it is completed, blocked or has exhausted its time quantum
(4) Roll-in/roll-out techniques (more than one program physically in main memory)

c. Multiprocessing

2. Scheduling

a. Data base integrity (protection from programs external to system):

(1) Operating system used to stop non-system accessing

b. Program scheduling and interrupt handling:

(1) Operating system function used
(2) Special scheduling within the DMS

c. Concurrency of operations:

(1) Concurrent operations on a data base may occur because:

(a) DMS allows more than one user to call simultaneously on the same or different functions
(b) Operating system allows more than one user to interact with the same copy of the DMS
(c) Operating system allows more than one copy of the DMS

(2) Concurrency during file creation:

(a) Two files can be created simultaneously
(b) The creation process on one file can be achieved at the same time as other functions on another file

(3) Concurrency with single copy (when only one copy of DMS exists):

(a) One application program called by two users who interact with different data files
(b) One application program interacting with the same data file, called concurrently by two users
(c) More than one application program interacting with different data files

(4) Concurrency with multiple copies (when more than one copy of the DMS is allowed within the operating system):

(a) Concurrent operations limited to one function, e.g., multiple concurrent querying of a file, but no concurrency of update
(b) Concurrent operations can only be performed on different data files
(c) Concurrent operations can be performed on both the same and different data files

3. Software Facilities

a. Use of operating system facilities by the DMS:

(1) Access methods (the basic I/O facilities of the operating system)
(2) Task scheduler
(3) Loader link editor
(4) Space and resource management (includes allocation of buffers and opening and closing of files)
(5) Communication facilities
(6) Additional features (sort programs, library utilities)
(7) Communications subsystems (terminal input and output control)
(8) Polling and queueing of messages is done by the operating system
(9) Major scheduling of users is the responsibility of the DMS

b. Other software:

(1) Compiler
(2) Sort/merge

4. Modes of System Use

a. Interactive mode:

(1) Prestored procedures (system does not actively assist the user):

(a) Execution required directly from terminal
(b) All necessary parameters supplied from terminal
(c) Capability is available to which DMS functions

b. Conversational mode:

(1) Capability is available to which DMS functions
(2) Scenario driven (walk-thru)
(3) User can request tutorial assistance at any time
(4) User can suppress tutorial assistance at any time
(5) Tutorials can be added
(6) Tutorials can be deleted
(7) Tutorials can be modified
(8) User can change from one level of dialogue to another at any time
(9) Specific tutorials provided for:

(a) File definition
(b) File creation
(c) File restructuring
(d) Language use
(e) Query formulation
(f) Output processing
(g) Error procedures
(h) Recovery procedures

(10) Acknowledgement:

(a) Acknowledgement of all user-initiated dialogue provided optionally at the user's request

5. Batched Processing

a. Multiple tasks

(1) Accumulation against a single file
(2) Accumulation against a given collection of files
(3) Processing of accumulated multiple tasks may be done in batch mode

b. Jobs may be entered through a remote terminal for:

(1) File generation
(2) Maintenance
(3) Retrieval
(4) Output

c. Jobs may be entered through a local terminal for:

(1) File generation
(2) Maintenance
(3) Retrieval
(4) Output

d. Composition of job request at a remote terminal
e. Composition of job request at a local terminal
f. Request library procedures from a remote terminal
g. Request library procedures from a local terminal
h. Parameters can be entered for a library procedure from a local terminal
i. Library procedures may be modified prior to execution:

(1) Temporarily
(2) Permanently

j. System informs user when:

(1) Job is accepted
(2) Job errors are encountered
(3) Job is terminated

k. Maximum number of active users that can operate simultaneously from remote terminals

6. DMS Transferability

a. System can be transferred between computers of the same family operating under the same and different versions of

operating system, or even an entirely different operating system

b. System can be transferred to operate on computers that are not in the same family
c. User must modify his procedures when transferring to different computers of the same family or a different computer family
d. User must transfer his data base from one type of storage device to another, either on the same or different computers or translate his data from one representation to another

D. Documentation Availability

1. System specifications

    a. System description (overview)
    b. Conceptual system description
    c. Detailed design specification
    d. Implementation specification

2. Operation documentation

    a. User:

        (1) File design guide
        (2) Language reference manual
        (3) System analysis guide

    b. System operation:

        (1) Operators manual
        (2) System maintenance
        (3) System administrative procedures

SECTION III

SOFTWARE TESTING ATTRIBUTES

1. INTRODUCTION

The initial question is: can a DMS be tested? The answer is yes: a DMS can be tested in several ways. In fact, its various functions can be measured in perhaps more ways than there are functions, which poses a significant problem in any DMS acceptance test or evaluation. What measurement technique or techniques would be most appropriate based on the specific user's requirements, objectives, and testing capabilities? Perhaps a simple listing of the attributes of a given DMS as implemented in a given computer environment will suffice. On the other hand, the modeling or simulation of implementation strategies may be employed so that performance data can be inferred or direct measurements obtained through the use of benchmark programs.

The three above stated techniques illustrate the two basic varieties of tests; those which are passive, i.e., do not involve the actual operation of the DMS, and those which are active and actually measure the system whether in whole or part. The passive variety of tests such as analyzing the attributes of various DMSs would constitute a soft measurement. The results would be qualitative based on the evaluator(s) judgment as to the degree to which system attributes or the total DMS satisfy user requirements. The active series of tests would quantitatively compare one DMS against another by measuring the time expended in performing certain specified functions (file maintenance and report generation, for example). These measures would be considered hard in that timings can be derived from the actual performance of the DMS.

This section, first, will discuss the importance of determining the criteria against which each DMS is to be measured. Then, the type of testing techniques, both active and passive, which can be used to measure the degree to which each DMS satisfies the criteria will be considered. Each technique, also, will be analyzed regarding its appropriateness and worth in the measurement of specific DMS attributes and a DMS in toto. Finally, a test methodology will be described that will serve as a guideline to DMS test personnel in test plan generation and execution.

This discussion of DMS testing and test methodology, however, will not attempt to evaluate the DMSs presently available. Evaluation implies a knowledge of specific user requirements, which are unique for each installation. It would be foolhardy to attempt to assign a value to each attribute, that would apply in each and every case. Instead, this paper suggests a methodology to measure and test DMSs and collect timing and performance data. The statistics, so gathered, then become the basis for making an evaluation. The collected data do not, however, of themselves, identify the best system since this depends on the user's particular requirements. A two step process is involved and this paper addresses the first step.

2. THE DMS EVALUATION PROCESS

Because Data Management Systems are unconstrained by any widely accepted standards, the task of DMS testing is hampered by the lack of standard measurement criteria. Therefore, in a specific operational environment, any measurement task involves a two-step process. First applicable criteria against which each DMS is to be measured must be determined. Then, specific measurement techniques are employed to determine how closely each candidate DMS satisfies the criteria. The type of criteria to be applied also may determine the type of measurement techniques used. Therefore, each step must be correctly done to ensure that a thorough and meaningful test program is devised.

a. Measurement Criteria

Measurement criteria can only be developed through some type of analysis effort. What requirements must be levied on a DMS to satisfy most the user's applications? This is not an easy question to answer because the conversion of user applications requirements to DMS requirements requires a thorough knowledge not only of the applications environment but also of DMSs in general. The selection of typical applications can be quite a problem in a multifaceted and complex operational environment. If there are diverse applications from which to choose, a subjective decision must be made as to which applications are the most typical and therefore can provide a baseline against which to judge DMSs. Even if the selection of the applications mix is done correctly, the next step, the tying of applications requirements to DMS functions is no easy matter. More importantly, until this step is taken, there are no specific criteria against which to judge a DMS.

Most DMSs possess the same general characteristics such as a file definition and creation capability, a procedural language to perform data retrieval and maintenance, and an output presentation package to edit, decode and format retrieved data for presentation. The problem of identifying these general characteristics is compounded by DMSs which rely in part on a multiplicity of implementation techniques. For example, some DMSs have differing procedural languages for maintenance and retrieval. Some have only a retrieval language and use table interpretation techniques for maintenance. Other DMSs are host language embedded and rely on the characteristics of the host language for data definition and maintenance and retrieval logic implementation; yet all of these variety of implementations are properly recognized as DMSs. Since many DMSs possess these same general capabilities, however, the analyst must look deeper into each function to locate attributes for comparison; for example, file organization, file overhead, access methods, etc.

If the user's prime requirement is rapid access, then he needs to consider the types of file organizations available from each DMS. This would be a major indicator of performance because file organizations dictate the search strategy invoked during data retrieval. Consideration must also

be given to the overhead requirement (pointers, indices) associated with each file organization. If storage space is limited, utilization of certain file organizations may be impossible.

The above discussion only points out the difficulty in relating data processing requirements to DMS requirements. Conflicts can arise between user needs and system resources and capabilities. Which file organization would best support the user applications--a ring structure, an inverted file, or a simple sequential file? What types of access methods should be evaluated?

In addition, the possibility that a DMS may not even be required should be explored. All classes of applications do not require such a system. For example, simple transaction files may only require some general list and maintenance capabilities that can be adequately handled by some simple COBOL programs, whereas other applications may cause maintenance or retrieval problems that are so complex that the capacities of a generalized DMS would be inadequate.

The determination of the utility of a DMS for a particular operational environment and the establishment of some measurement criteria with which to evaluate the varied DMSs is a most important task. If it is not performed, there would be no standard guidelines to use during the actual testing of the systems, and if it is poorly done, the validity of the subsequent tests is questionable.

b. Measurement Techniques

Measurement techniques can be grouped into two general categories. active techniques and passive techniques with benchmark testing, modeling/simulation and monitoring residing in the former category and analysis and numerical scoring the latter.

Each of these measurement techniques provides test personnel with some data regarding system capabilities and/or system performance. The data will vary depending on the technique used; therefore, the selection of a particular technique(s) must be based on the measurement criteria that had been determined during a preceding analysis. In other words, determine what you are trying to test and then select the technique(s) that provide results that can be applied against the measurement criteria previously selected. Section IV presents a matrix which associates DMS attributes with specific testing techniques which can be of use for both test planning and execution. Test personnel would know the type of techniques required to fully measure the systems. This would provide some lead time to establish a capability in a particular area (for example, software monitors) if one does not already exist. A cost figure to conduct the tests also can be estimated and compared against the value of the anticipated results. The following presentation provides a description and evaluation of each technique.

(1) Active Testing Techniques

(a) Benchmark Testing

Benchmarks often have been employed as a measurement technique for the testing of computer systems. Benchmark programs are a mix or grouping of actual or live applications to be executed by candidate DMSs in order to obtain comparative performance figures on their capabilities to handle the various applications. This mixture of applications usually reflects a percentage of the total work load, the execution of which would normally be prohibitive from a resource utilization standpoint. Therefore, each benchmark program should correspond "percentage-wise" to the amount of processing time required of the application it represents, to insure that the benchmark testing is representative of the actual processing environment. A standard operating environment is assumed, and the main output of this technique is, typically, sets of timing measurements.

The key characteristics of benchmark testing are threefold. First, the programs are actually run under the candidate DMSs. Secondly, the total throughput time is important (not just the processor time) and finally the programs are aimed at specific applications that are hopefully representative of the operational environment.

They might be best referred to as a "real" simulation but with no formal data acquisition and reduction methodologies. Two distinct approaches to evaluation can be provided by benchmarks. One approach called "Kernel Timing Estimates" is used to measure central processor timings, while the other "Benchmarks Problem Timings" are used to evaluate the entire computer system.

i. Kernel Timing Estimates

Kernel analysis attempts to evaluate software systems by comparing the time (and costs) required to perform a specific function. In this technique, the central processor time utilized during the DMS function is the measurement tool and the code generated by the CPU in performing the function is called a "kernel". In a DMS evaluation, candidate functions would be 1) update, 2) generate, and 3) retrieve. The efficiency of these functions, measured in CPU time, can then be ascertained by utilizing application programs that initiate the function execution. Within a DMS, the various strong and weak points of a system can be determined by using the kernel approach. By the same token, the kernel approach can also be used to compare one DMS against another. However, to make the comparison meaningful, some sort of weighing technique is required. Needless to say, the collection and use of such weights involve problems similar to those associated with any type of instruction mix problem. Moreover, the problem of sub-optimization must be assumed at some level. Consider the retrieval and update functions. The cost of the former in terms of the latter is certainly less costly in an inverted file organization with a high volume of traffic than in sequential file organization with a high volume of traffic. Any single set of weights must thus represent sub-optimal use of one (or both) systems.

Note, however, that the situation is not that hopeless; kernel analysis is only meant to be an initial step in the testing of a DMS, not the final evaluation.

ii. Benchmark Timing Estimates

The second approach to benchmarks or "benchmark program timings", does not concentrate on CPU timings but on system timings.

The main advantage associated with benchmark testing is that "typical" applications are being run on the proposed system and meaningful measurements of system running time can be obtained.

These timing measurements, however, are only as good as the benchmark programs. If the latter are not representative of typical applications or percentage-wise do not reflect the normal operational load, then the test results are questionable and, in many cases, worse than no test at all, since otherwise good systems may be judged negatively. Benchmarks also do not necessarily pinpoint the problems associated with a particular DMS. Poor throughput might be indicative of poor DMS-OS interface rather than just the DMS. Even if the problems are within the DMS, it is difficult, if not impossible, to determine their location using benchmarks. If the testing is concerned with judging only the DMS and not the total system, then benchmarks of and by themselves are not adequate.

An additional problem concerns the standardization of benchmark programs. If each manufacturer codes the programs to operate under their particular DMS, then the speed of system throughput may be more a function of the quality of the programming than the DMS.

iii. Conclusion

Benchmark programs are a useful tool in the testing of an overall system configuration of which the DMS is a part. There are associated problems, however, which must be considered prior to arriving at a decision regarding the functional efficiency of a particular DMS within an operational environment.

It must be noted that in any benchmark test of a DMS on a third-generation multi-programming computer system, it is not possible to test only the DMS. Because of program relocatability and the speed and concurrency of system functions, various software modules cannot be isolated for kernel analysis, etc. The derived timing data applies to the system as a whole including the hardware and all software (operating system, DMS and applications programs). In this case, then, it is not accurate to state that the benchmark is testing a DMS. It is testing a whole system of which the DMS is one part.

(b) Modeling and Simulation

i. Introduction

Many modeling techniques have been developed in recent years to study and optimize storage structures, search strategies, and device usage. Most attempts at modeling are directed at specific applications and at optimizing subsystems or device utilization within a DMS.

Whereas a model reflects a particular state of the system at a fixed point in time, simulation or functional modeling is usually thought of as a demonstration of artificial performance involving a dimension of time. Consequently, a simulative study will usually include a model of some sort which is exercised in such a way as to produce a continuum of states reflecting overall performance. In this sense, then, simulation is an abstraction of a system that can be used to predict real behavior. This is done by describing a series of experiments representing variations of the parameters of a behavioral model.

Before selecting an evaluation approach, careful consideration must be given to the desired goal and the environment constraining the system of interest. Analytic approaches to evaluation may be developed for application either before-the-fact or after; either for system design/improvement or for evaluation and measurement. This distinction can be clarified by considering the conditions under which the analysis is applied. For anticipated systems, a reliable prediction of expected performance is desirable. Likewise, for extant systems one might develop a method of predicting performance in a new environment, with different parameters, or under a modified implementation scheme. Prior-use applications of the predictive approach are in some cases similar to but still distinct from analytic techniques for a posteriori evaluation or measurement. In either case, the ultimate goal of optimum performance is the same.

In choosing a modeling or simulation technique, the following questions must be answered:

- Are we reasonably certain that we can obtain either an exact solution or a satisfactory approximation to the solution of our problem by making use of a given tool?
- Is this the lowest cost computational procedure for solving our problem?
- Does the particular technique under consideration lend itself to relatively easy interpretation by those who are likely to use the results of the study?

Of course it is paramount to the consideration of these questions that the problem be properly identified and well-defined. But this is a necessary step in any case. Subsequently, a model

must be formulated which validly reflects (i.e., "models") the states of the system of interest. The structure and flexibility of this model determines whether the investigation can be expanded into a meaningful simulation

There are then three phases in the investigation that remain after the problem has been identified and a model formulated:

- Model Implementation - The model must be described for use in a particular installation. At least four different methods have been associated with this step.

  - Use of a "packaged" model such as FORMS or SCERT for which only the input parameters need be specified.
  - Use of a general purpose language such as COBOL or JOVIAL. This approach is often used for smaller scale, locally designed models.
  - Use of special language or routines in conjunction with a standard algebraic language: e.g., SIMTRAN, GPSS.
  - Use of specially designed simulation languages expressly intended for modeling and simulation such as SIMSCRIPT and GPSS. This may be even more specific in that the special language may also be oriented to a specific function or subsystem type.

  When a choice exists, there are several factors to consider in selecting a simulation language. Among these are ease of learning, expressiveness (the ease with which the model can be described in the language), compilation and execution speeds, reporting facilities, general computation capability, and execution time facilities. The relative importance of these considerations depends upon the problems at hand. If the requirement is to build a number of different small- to medium-scale simulation models, GPSS may prove most efficient; for large models or models in which much general computation is required, SIMSCRIPT may be preferred

- Strategic Planning - Design of an experiment that will yield the desired information. This is not an easy task - it involves the careful design of the environment to be considered by the model.
- Tactical Planning - Determination of how each of the test runs specified in the experimental design is to be executed. Again, care must be exercised in the selection and construction of input parameters to the model which will allow it to reflect the desired performance.

ii. Utilization

The basic procedure to be followed in the utilization of this technique is given in nine steps. (Note the relationships indicated between model - steps 1 to 7 - and simulation - steps 8 and 9.)

1. Formulation of the problem
2. Collection and processing of real world data
3. Formulation of mathematical model consisting of components, variables, parameters, and functional relationships
4. Estimation of parameters of operating characteristics from real world data
5. Evaluation of the model and parameter estimates
6. Formulation of a computer program to realize the model
7. Validation of the model
8. Design of simulation experiments
9. Analysis of simulation data (performance)

Steps 1 and 2 of this process are discussed under Measurement Criteria above. Data for step 4 may come from Analysis, Benchmarking, etc. The basic model development then is given in steps 3 and 5-7. These comprise the main bulk of a simulative development. Hence, the use of "packaged" systems, where applicable, saves a lot of effort. These steps are then replaced by that of finding and choosing an appropriate simulator.

One such system worthy of note is the File Organization Modeling System (FORMS) whose development is being supported by RADC. This system was extended recently by PRC to handle additional devices and search strategies. The resulting model provides RADC with a

more generalized capability to model selected functions of data management systems on the specific host environments simulated by FORMS.

This model permits the analyst to evaluate various file organizations, device types, and access methods. He describes his data file in terms of record size, count and blocking factor, device location, and record distribution and then tests various retrieval strategies against specified file organizations to determine the most efficient organization and access method. With each test, processing time, space utilization, avarage access time and file I/O activity are provided on an execution report. File organization and access method can be varied to provide comparison data and performance curves depending on the particular file organization and access methods utilized.

The utilization of such methodology, however, is necessarily restrictive. The presently available models either concentrate on a limited range of DMS functions such as file organization and data access methods or are too general to be of detailed use in the evaluation of DMS functions. The information provided is useful mostly from the standpoint of maximizing an already implemented DMS or in determining general guidelines for the development of a new system.

When a simulative model must be developed from scratch, one of the most difficult tasks is the validation or "calibration" of the model (step 7 above). There are three basic ways to validate:

- o Compare simulated results to actual performance of a realized design. This, of course, is only possible when the system simulated already exists somewhere else - this is very improbable since most simulators are developed to aid in the design of new (non-existent) systems.
- o Compare simulated results to those of another (similar) model. Of course the evaluation is only as valid as the models which are being compared.
- o The third scheme is not as straightforward as the other two. It involves a fairly detailed knowledge of the model and the environment to which it applies. The basis of this scheme lies in the fact that most models can be partitioned or artifically exercised to produce (partial) results which can be verified by hand calculations, observation, or comparison to known physical characteristics of the environment. Use of this method might be considered validation by parts.

It may be evident at this point that model validation reduces to basically the same problems that must be faced in the DMS evaluation to begin with.

There are many techniques used in the development of simulation models. Depending on the system to be modeled, these may include any combination of methods from applied mathematics and logic including statistical theory (stochastic - probabilistic, or deterministic processes), queueing theory ("waiting lines"), and linear programming (for objective function optimization). An example is presented here which illustrates how one of these techniques can be used to evaluate performance in a DMS.

One method of providing relatively rapid random access to sequential files is by the construction of a hierarchy of indices (indexed sequential organization). With this arrangement applied to disc resident files, there are a minimum of two disc accesses required to locate a data record - one to get the cylinder index and one to get the track index. If it were possible to predict the record location from the cylinder index alone, one of the accesses could be eliminated at a considerable saving in time. It need not even be necessary to always predict the exact location. In other words, a scheme which periodically eliminates one access may still be beneficial. The problem then is, can access to the track index be replaced by evaluating a predictor function of the cylinder index? One must be aware of the possibility that a predicted value which is incorrect then causes an additional access. Applying this scheme across a full DMS application then, one can rephrase the problem as: what is the average number of redundant accesses generated if the data location is predicted? A statistical model of the environment must be formulated to solve this problem. The following data is pertinent: the number of disc tracks used, the number of keys (and records) and their statistical distribution across the tracks, the density of the keys within tracks, and the search strategy undertaken as the result of a query for a specific key. The cylinder index must then be interpolated to find a neighborhood of tracks in which the desired key lies. This neighborhood must be small enough so that fewer than two accesses are required on the average. The complex statistical model developed might show that an average of, say 1.9 accesses per record are required. This may not be cost-effective at all considering the extra software that would be needed to run the predictor for each query presented. On the other hand, an average of 1.1 accesses may improve over-all execution time considerably. By applying variations of file size, record distribution, and direct access device characteristics, one may determine the feasibility of using this modified hierarchical indexing scheme in a given environment.

iii. Conclusion

The cost of developing and operating a comprehensive Data Management System Simulator capable of supporting study of all aspects of many DMSs in a variety of environments can easily become prohibitive. Hence, care should be exercised in selecting this approach. The benefits of simulation are best realized when the system being simulated is very large, complex, and too expensive to evaluate by any other means.

Since models are related to particular states of a system, they are generally more parametrically flexible which can lead to a better understanding of (sub-) system interactions. It is when these interactions reach complexities not readily comprehensible that a simulation may be necessary to demonstrate trends in behavior and performance.

It should be emphasized that simulation is not a panacea for arriving at successful implementation or accurate performance prediction and evaluation. Although the recent popularity of simulation lends support to this idea, sufficient counter-examples have been documented to refute this position (1). Sufficient valid results often can be obtained by statistical formulations or other models alone.

The extent of the procedures involved in simulation and modeling are what make this approach expensive and difficult. The factors and algorithms used are often difficult to derive even for existing systems let alone prospective ones. In addition, describing the functions to be performed or the number of times each function is to be executed requires a knowledge of the applications that most users do not possess. Furthermore, the development of the simulation model is only a part of the analysis process. Subsequent steps include testing and validation of the model as well as the design of experiments, and the analysis and interpretation of results.

(c) Monitoring

A computer-aided measurement method is the collection of the statistics and actual performance parameters of an operating, live system. Monitors are built within programs and within systems, and they can also be external to the system being monitored. There are two general classifications of monitors presently in use: hardware and software.

i. Hardware Monitors

A hardware monitor obtains signals from the computer system by attaching directly to the computer's circuitry high-impedence probes which measure the presence or absence of electrical impulses. The monitor is basically a set of counters which record the occurrence of certain significant events such as CPU and channel activity. Performance figures are obtained by measuring the number of impulses and the duration of time between given events or the logical combination of the events of a computer system.

The counters usually have two modes of operation: a time mode and a count mode. In the former, the counter accumulates pulses from the monitor's internal clock upon receipt of an active signal from the computer circuitry being tapped, i.e., CPU, channel, device. The monitor continues to accumulate pulses until the signal stops. Since the clock frequency and the time value of each pulse are known, the total

overlap time is obtained by multiplying the value recorded in the counter by the time-value/increment. The result will show how long a part of the system was in use during a specified time frame. In the count mode, the count simply is incremented each time the circuit being monitored goes from an inactive to active state. It indicates the number of times a part of the system was used during a specified time frame.

The primary advantage associated with the utilization of hardware monitors is that such devices can function without utilizing any of the host computer system's resources. The monitor neither degrades nor interferes with the system that it is monitoring, and the data collected is a fairly accurate representation of system performance. The disadvantage is that the monitor cannot dynamically evaluate the measurement being taken to determine if, in fact, it should be included in the system test.

Hardware monitors range in complexity from the very simple basic counter unit with manual readout to full minicomputer systems. Because of this difference in capability, the following points require consideration prior to employing a particular hardware monitor as a measurement tool.

(i) Speed/Timing

Speed and timing requirements are dictated by the system to be measured. The monitor must be capable of detecting the smallest signal from the host computer. For example, if the smallest signal that a monitor can detect has a minimum pulse width of 300 nanoseconds and if its repetition rate is one million counts per second (1-MHz), whereas the system it is attempting to monitor had a minimum pulse width of 100 nanoseconds at 3-MHz, the collected data would be useless, because the monitor would fail to recognize and thus not count any pulse under 300 nanoseconds wide. In addition, the clock also must be fast enough to provide good resolution, otherwise inaccuracies can creep into the collected data.

(ii) Number of Counters

Any monitor should have at least eight to ten counters. Each counter, also, should be capable of performing certain logic functions (AND, NAND, OR, XOR, NOR, INVERT).

(iii) Probes

The probes must be compatible with the computer system to be measured and caution must be exercised to insure that they place no drain on the computer.

(iv) Compar

A comparator is a device which measures the amount of time spent in a certain section of the code. The probes

are connected to the instruction address register and the address range on the comparator indicates the starting and ending address of the code segment to be measured. The address of each code segment brought into core is then compared against the address range in the comparator, and if they are a match, the amount of time the code segment is active is calculated.

This device can be quite useful in measuring the amount of time spent within particular DMS modules, for example, retrieve and/or store. A problem, however, arised in third-generation multi-programming systems where software modules may be relocated within core to optimize space usage.

(v) Technical Support

Unless the installation to be monitored has a resident systems engineer who is familiar with the hardware backboard wiring schemes, adequate technical support must be available to determine, based on what is to be measured, where to attach the probes. Without this knowledge, the monitor is useless.

Data reduction and analysis routines also should be provided as part of the technical support package because of the voluminous amounts of data that can be generated.

The above discussion illustrates the capabilities and requirements that should be part of any hardware monitor. The mere presence of a monitor, however, does not mean that system testing can begin. First, the user must determine what it is he wishes to measure. Will the knowledge of CPU activity, CPU-I/O overlap, application program versus OS time and/or activity by region help in system evaluation? The answer, of course, is "yes", but only if the evaluator has a detailed knowledge of the total system including the OS, DMS, application programs, and user data and files. Even if this requirement is fulfilled, data interpretation and the lack of specificity are problems that remain. Data, regardless of the amount, requires interpretation to be of use in an evaluation task. The speed of third generation machines results in so much data that it is not unthinkable that the evaluator could be overwhelmed. Also, most DMSs operate in a third generation multi-programming environment where the volume, speed and concurrency of operations make it most difficult to focus on specific DMS functions. To extricate the DMS from this environment would only invalidate the measurements performed. Therefore, in developing a DMS test methodology, the use of hardware monitors has a limited applicability.

ii. Software Monitors

Software monitors are actual code segments embedded in either the DMS or operating system to collect performance data on specific DMS functions and/or the total DMS. Calls to a subroutine can be embedded within specific DMS functions such as retrieve, store, etc., to collect timing data on particular functions.

The scheduling of events and the time interval between events also can be determined. Other functions that can be monitored include the user program - DMS interface, the DMS-OS interface, the access methods, and the efficiency of the DMS procedural language. It should be noted here that the procedural language has a predominate influence on the power and efficiency of the DMS. For example, COBOL provides IDS with all the capabilities possessed by the host language including all the data manipulative and formatting functions, whereas other more structured DMS languages often provide only the more basic capabilities. Therefore, in any DMS test exercise, close scrutiny should be given to the language.

Software monitors also can be used to identify bottlenecks in the system. For example, if the system often is waiting completion of an I/O function, perhaps the reallocation of particular peripherals could alleviate the problem.

The utilization of this technique presumes a familiarity with the DMS, and if this is not the case, then adequate documentation of the system must be available. Otherwise, it would be impossible to properly insert calls within specific segments of the DMS. This situation is mentioned in the DMS Test Methodology Validation document, where, because of inadequate documentation, software monitors could not be embedded within the ADVISOR system. System flow charts can serve as an excellent guide for identifying those DMS functions test personnel wish to monitor. Once the various subroutines have been identified, the DMS source listings can be used to pinpoint the exact spots to embed the calls. However, if system flow charts do not exist or the source listings are not commented properly, it becomes a most difficult task to identify those locations in which calls should be embedded.

One advantage associated with the use of monitors embedded within the DMS is that less code is required to collect the performance statistics. A small routine can be written to access the system clock at the start and conclusion of the monitored DMS subroutine, and then format and write the output records. Even if the event sequence and an occurrence count are maintained, the size is not dramatically increased. The only other software required would be the calls themselves embedded at the beginning and end of the monitored subroutine. The insertion of these calls must be done carefully, however, to insure that all branches within the subroutine are identified and covered.

In addition to the insertion of monitors within DMS modules, monitors also can reside as separate programs within the operating system. These monitors can be generated locally to collect specific data regarding the DMS; or the system accounting program which is generally available at most computer installations, can be used. In the former case, an executive monitor would supervise all system activity and when certain modules were called, the monitor would be initialized by the operating system and collect the specified performance data. This procedure, however, is complicated by the fact that the monitor must be small

enough to be able, generally, to reside in memory and still have the capability to determine system status and module activity. This, however, requires access to the system tables which, in turn, increases the amount of code and, therefore, the overall size of the monitor.

System accounting programs, on the other hand, are generally already part of the operating system and special programming is not required. However, because they are general in nature, only recording the computer resources used in processing jobs, they lack the specificity to be of great use in the testing of DMSs. The data collected corresponds in many ways to the data obtained through the use of hardware monitors (CPU time, number of I/O accesses, core size and start/stop times).

The collection of such generalized data, because of its quantity, creates a data reduction, collation and analysis problem which must be considered when using any type of monitor. The mere presence of the data does not provide any answers. Certainly, the utilization of an inverted index will provide faster response than a serial search of a data base but what price in terms of storage overhead and generation and maintenance time must be paid for the index, and is it worth it? The utilization of regression and cluster analysis methods has been used to measure the effect of a system modification on the performance of a total computer system, and such methods may also be pertinent to the measurement of DMSs, but more study in this area is required to determine the independent and dependent variables that should be used and the accuracy of these methods. Such methods and the use of accounting data in computer performance analysis is fully explained in the Rand publication, Computer Performance Analysis· Applications of Accounting Data by R. Watson (2).

Regardless of the location of the software monitors within the DMS or OS, they, unlike hardware monitors, do utilize system resources, because of their own requirements for core, peripherals, and, above all, time. This is not an important consideration in regard to monitors associated with obtaining system accounting data because these are already part of the operating system. However, other software monitors would degrade system performance because of their own requirements for resources and this degradation should be measured. This is not a simple exercise, however, for the calculation of the time spent in monitoring particular DMS functions also utilizes system resources and therefore degrades the system even more. Test personnel must take this into consideration when analyzing the results obtained from such techniques. In addition, the precision of any measurement can be no greater than that of the accessible timer in the host computer.

The specific advantage that software monitors possess over hardware monitors relates to their flexibility and range of capabilities. They can be inserted anywhere within the system and because they reside in memory, they have access to all system tables and can measure any and all aspects of the system including core usage, queue lengths, data access speed and individual program/subroutine operation. Their flexibility and usefulness is clearly indicated in the DMS Test Methodology

Validation paper produced by PRC in conjunction with this document. Software monitors were used to measure the timing and performance of the MADAPS system. Because of the availability of adequate documentation, software monitors could be embedded within specific DMS functions and timing and performance data obtained. The technique was neither time-consuming nor difficult to implement and the collected data, upon analysis, provided some clear-cut insights into the functioning of this particular DMS.

Another advantage associated with the use of software monitors is that once generated, they can be used again and again to measure system performance. A slow degradation of the DMS could occur, over time, because of an increase in transactions, file size or new applications. The periodic insertion of software monitors within the DMS could identify such problems and give the data base manager some lead-time to find and implement an acceptable solution.

Another approach worth consideration is the use of hardware monitors in conjunction with software monitors. This methodology would require fewer code segments within the DMS and/or OS, and therefore, would alleviate, to some degree the overhead associated with software monitors. By synchronization of the clocks in both the hardware monitor and computer system, channel, CPU and device activity can be compared with module activity to determine the overall efficiency of the system. Are there I/O tieups due to poor module/channel allocation? What causes the wait states in the system? Does the DMS degrade the system?

Monitors have the potential to become a very useful tool in the measurement of DMS performance. Their utilization will provide comparison data with which a skilled analyst can accurately evaluate a DMS. Monitors are not easy to implement but the results yielded by this method can be well worth the effort.

(2) Passive Testing Techniques

Analysis and numerical scoring are the two most commonly used techniques that can be considered passive. They are typically manual methods whereby DMS parameters, having been determined, are rated in degree of importance. The parameters can be considered as a whole with each possessing, relatively, the same degree of importance (analysis) or they can be ranked (numerical scoring).

(a) Analysis

Analysis techniques provide computer-aided studies of system performance in which the actual capabilities of the systems in operation are studied. A thorough knowledge of user requirements already should have been determined, and these requirements transferred into general DMS functional capabilities prior to initiation of systems analysis. For example, the requirements, if any, for a remote batch and/or on-line capability need to be determined. Also, the general capabilities required in the DMS procedural language must be noted. Then, DMS systems documentation

can be reviewed, the manufacturer questioned, and those systems that fulfill the listed requirements selected. If the systems are operational, their general capabilities can be analyzed in an actual processing environment. The programmers and analysts who interface with the DMS can be interviewed and system performance studied during normal processing. The general capabilities of the system, file definition, maintenance and retrieval, can then be documented with the various pros and cons listed.

There are, however, numerous weak points in this approach. It is much too general to be anything but a first step in any test methodology, and even as a first step, it is lacking. Since many of the more prominent DMSs offer the same general capabilities, a large number of systems that should be eliminated are not. Also, where system problems are observed, the analyst has no way of knowing whether the DMS, the operating system or poor application programming is at fault. Interviews to pinpoint the problem could be misleading and in many cases would prove to be inconclusive. Therefore, unless the DMS also could be observed in a relatively smooth operating environment, erroneous judgments might be made. Finally, this method is incapable of evaluating a grouping of DMSs in which the system capabilities of each vary in their support of diverse applications.

(b) Numerical Scoring Methods

This evaluation technique is typically a manual method whereby parameters of various systems are developed and assigned a numerical rating. The higher the score earned by a system, the better that system's general performance. Chief among these methods is the Parametric Evaluation of Generalized System (PEGS) (3).

The approach is to establish a set of DMS requirements (parameter list) based on user requirements/applications and evaluate the capabilities of the applicant DMSs against the selected parameters. These parameters are scaled and assigned weighted values based on their relative priority within the operational environment. The capabilities of the candidate DMSs then are compared against each parameter and a rating assigned. When each DMS has been rated, overall scores are computed.

This methodology serves two important functions. It establishes a set of criteria against which the candidate DMSs can be judged. And it ranks the systems so that those with the poorest overall ranking can be eliminated from further consideration.

The drawbacks associated with this technique are that a heavy reliance must be placed on system documentation which often is quite misleading. Certain capabilities may be much harder to implement under one DMS than another, but this certainly would not be revealed in the manufacturer's documentation. Additional information may be obtained by visiting a site where each DMS is operational, assuming this is possible, but again, other variables come into play which could create a false impression of overall system capabilities.

A more severe problem associated with the utilization of this technique, however, is the difficulty in generating a parameter list based on the application requirements and the assignment of weighted values to each parameter. This problem becomes even more difficult of solution when widely diverse applications are part of the operational environment. Very specific relationships between the application requirements and DMS functions must be devised. What DMS capability will best satisfy the requirement? This type of task requires personnel with a wide range of expertise in data management in addition to a total familiarity with the applications environment - a capability which is not prevalent in many installations.

The results from such a technique, therefore, must be qualified because of its inherent problems. As in the discussion of the analysis technique, both serve as a logical starting point in any evaluation process, but more sophisticated techniques must later be applied to adequately evaluate a DMS.

3. DMS TEST METHODOLOGY

The preceding examination of measurement criteria and techniques has illustrated the various methods presently in use to test Data Management Systems. The following section will propose a generalized DMS Test Methodology to be employed in utilizing the aforementioned techniques. The methodology will be structured so as to permit the utmost flexibility in solving the widely diverse DMS measurement problems that arise in the present day operational environment such as the evaluation and selection of a DMS, acceptance testing an already selected DMS and even in the identification of problem areas in an operational system. This methodology also could assist management personnel prior to the actual testing by providing them with a frame of reference for the development of a test plan. Before any testing program is initiated, management will wish to know projected figures on the level of effort required to test the DMSs. This methodology, in conjunction with the matrix presented in Section IV, can indicate the potential areas of activity and from this, resource requirements can be estimated. Thus, the proposed methodology can be of service in both the preparation and actual execution of a test plan with the decision as to its use lying in the hands of each individual user.

This paper does not suggest that the proposed methodology is the only way to solve a DMS measurement problem. This would indeed by foolhardy because of the indeterminable number of variables that can apply to any one situation. What is suggested, however, is that the methodology serve as a guideline in the measurement process, providing a step by step approach to the most common DMS measurement problems.

This section consists of two parts. Part one describes the preparatory analytical steps in the DMS test process while part two addresses the active and passive techniques and their utilization. Figure 1 graphically portrays the proposed DMS Test Methodology. As can be seen from the illustration, the methodology is both multi-entrant and multi-exited which permits the

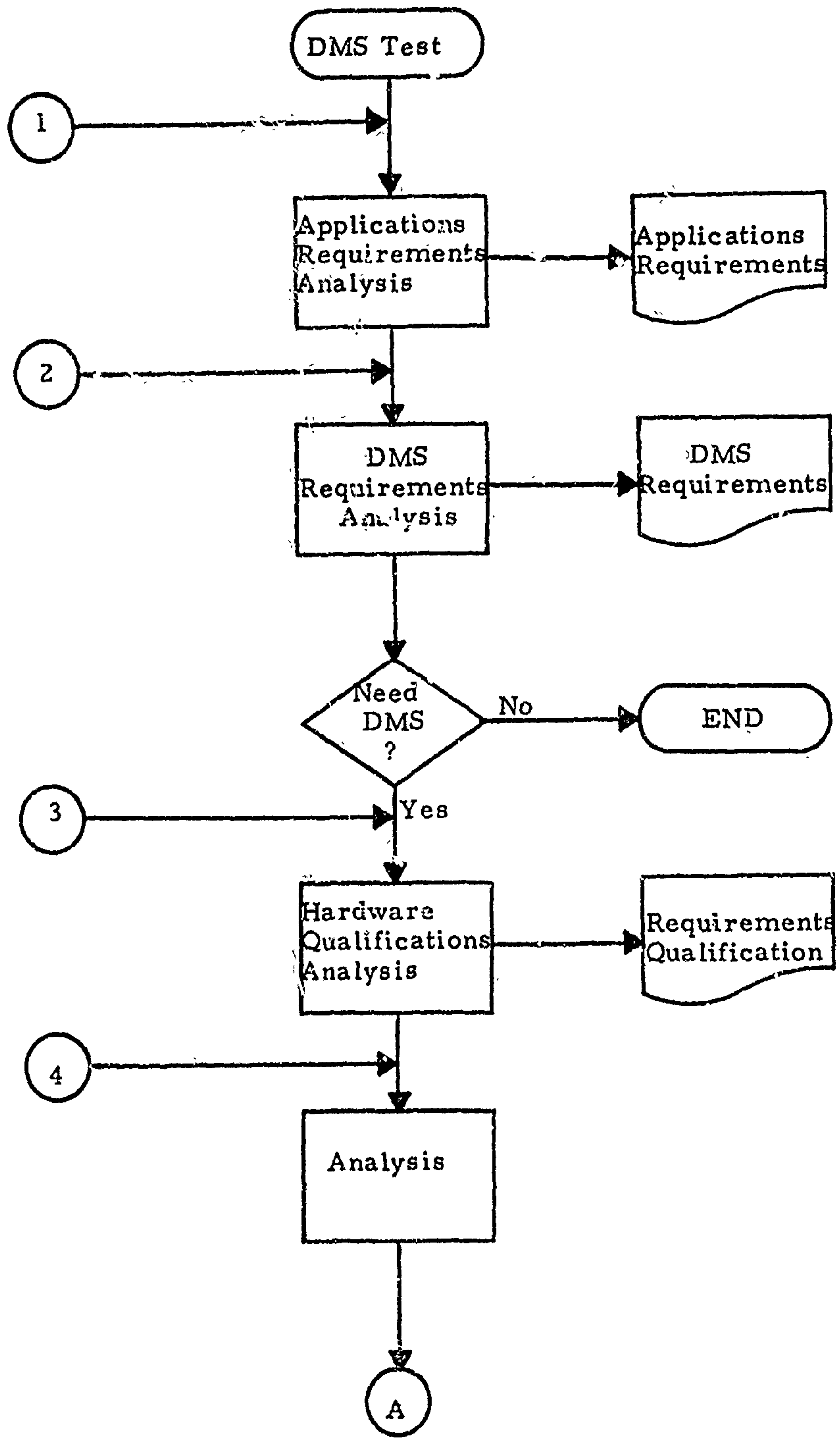

Figure 1. DMS Test Methodology

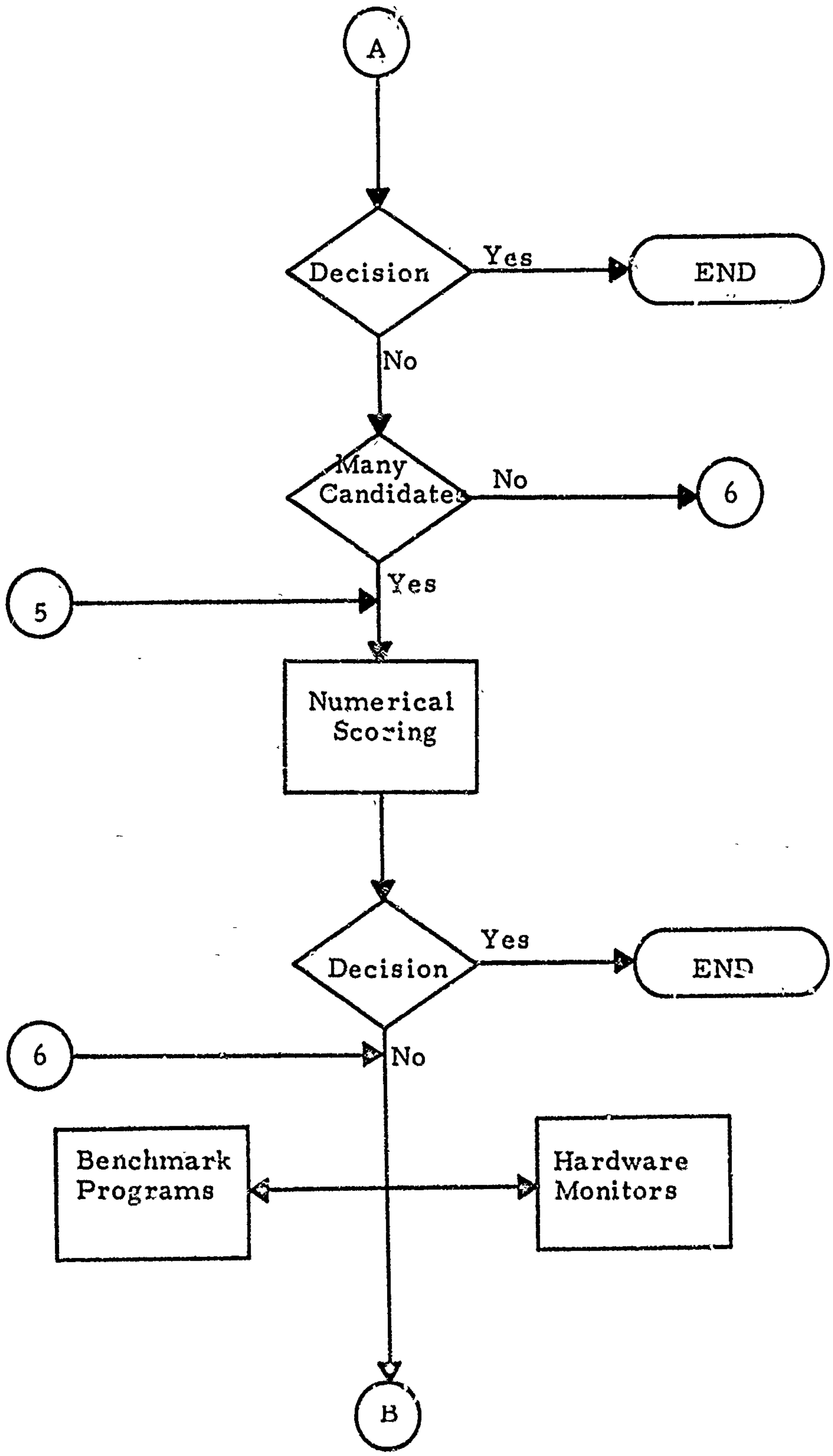

Figure 1---Continued

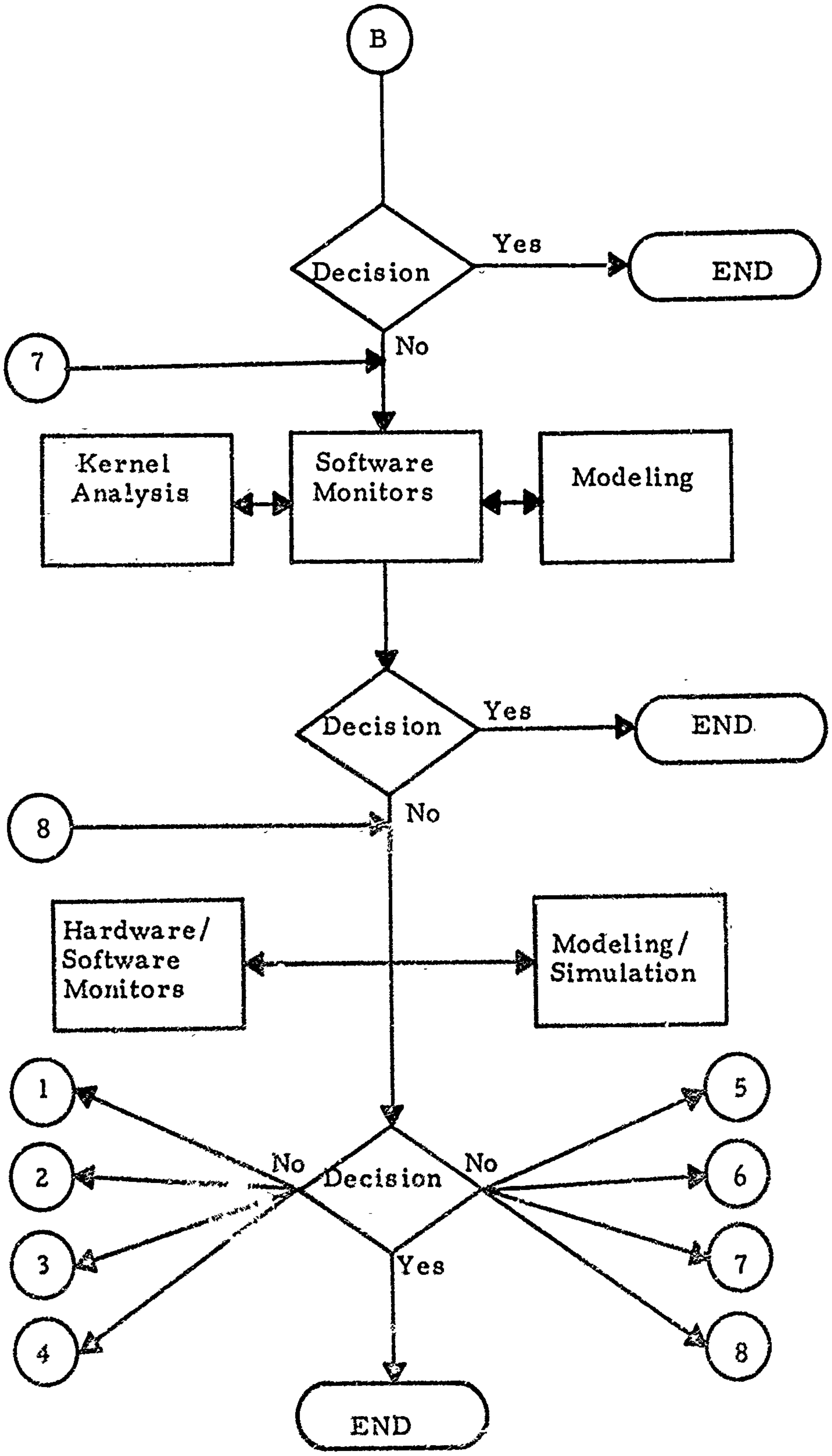

Figure 1---Concluded

methodology to address the many types of DMS measurement problems. This trait and the flexibility associated with it results in a multi-purposed and generalized DMS Test Methodology that is capable of addressing the majority of situations involving the selection and/or optimization of a DMS.

a. Preparatory Steps

The DMS Test Methodology consists of three preparatory steps; 1) perform an analysis of the applications requirements, 2) convert these applications requirements into DMS requirements, and 3) qualify the DMS requirements based on an analysis of the hardware environment. These steps perform two important functions; 1) determine whether a DMS is needed to solve the operational problem and 2) if a DMS is required, to determine the criteria that should be employed in testing the candidate systems.

(1) Step 1

The first preparatory step calls for an analysis of the user's applications requirements. What is the user attempting to accomplish? This step might seem rather straightforward at first glance, but diverse applications requirements can add complexity to the process. Only by understanding the user's requirements can the individual or group responsible for the test proceed to Step 2 where these requirements are converted into DMS specifications. This conversion process requires that the test personnel be familiar with both the user requirements and data management systems in general for the procedure of specifying the particular DMS functional capabilities that will best satisfy the user requirements is no easy task, particularly when diverse user requirements exist. The test personnel must decide what are the typical or critical applications that would require servicing from a DMS. It may even be necessary to rank the applications in order of importance to facilitate the establishment of the most important DMS requirements.

(2) Step 2

Step 2 requires that the applications requirements established in Step 1 be converted to DMS requirements. This step, in effect, establishes the DMS criteria that will be used to judge those DMSs deemed worthy of consideration, and is, therefore, vital to any DMS test process. This step requires that the test personnel possess a thorough knowledge of both the applications requirements and DMSs in general because the process of matching an applications requirement with a particular DMS characteristic is not a straightforward procedure. For instance, what DMS trait will best satisfy a user's requirement for fast response? In fact, the test personnel may even conclude that a DMS would not best serve the applications requirements and propose that other software be utilized. This situation could arise when the applications are very limited and basic so that a simple COBOL program would suffice, or when the applications may cause maintenance and retrieval problems that are so complex that they are outside the capabilities of a generalized DMS. If this conclusion is reached, then the test is terminated. If not, then Step 3 should be performed and the results of the conversion from user to DMS requirements can be

considered in light of the test pairing accomplished in Section IV to indicate the type of techniques that can subsequently be utilized in the measurement of the desired attributes. The test pairing matrix will also indicate how deeply into the hierarchy the test process will have to proceed to satisfactorily test the identified attributes. This can be of great value in estimating the cost and time that will be required to adequately test the DMSs.

(3) Step 3

The set of DMS criteria developed in Step 2 is now considered in light of the hardware environment, and any needed qualification of the criteria is accomplished.

If the DMS to be selected is to run on an already operational system, then the type of equipment has a profound effect on the DMS selected. For example, IDS which is a host-contained DMS, only can execute on selected Honeywell (formerly GE) configurations. Therefore, if IBM equipment is used, IDS is already eliminated as a candidate system.

The decision whether to select a host or self-contained DMS also must be qualified by the hardware environment, since host systems presently are tied to a particular hardware configuration, and, therefore, a number of qualified systems may be eliminated for the same reason as given above.

However, if the hardware configuration is being selected in conjunction with a DMS and there are no strong reasons to select one hardware manufacturer over another, then the testing of both equipment and DMS can be intertwined and ultimately lead to the selection of the best combination of hardware and software. This, in effect, would result in the selection of the best overall system in terms of the DMS-associated applications requirements since certain qualified DMSs would not be disqualified simply because of the previously selected equipment.

If any or all of the above steps had already been accomplished, then of course they could be omitted. Once the requirements for the first three steps have been fulfilled, the utilization of the active and passive techniques can begin.

b. Active/Passive Technique Utilization

The DMS testing process resembles the pyramid structure depicted in Figure 2. It is hierarchical in that you proceed from one technique to another as the testing becomes more specific. The techniques at the tope of the hierarchy, analysis and numerical scoring are inexpensive and relatively easy to employ. They are used as a filter to trap the majority of DMSs and only let the most qualified pass through. Those that pass through would then be tested using the more expensive and difficult techniques such as benchmarks, monitors, modeling and simulation, and since only a few DMSs would be so tested, their utilization would be practical.

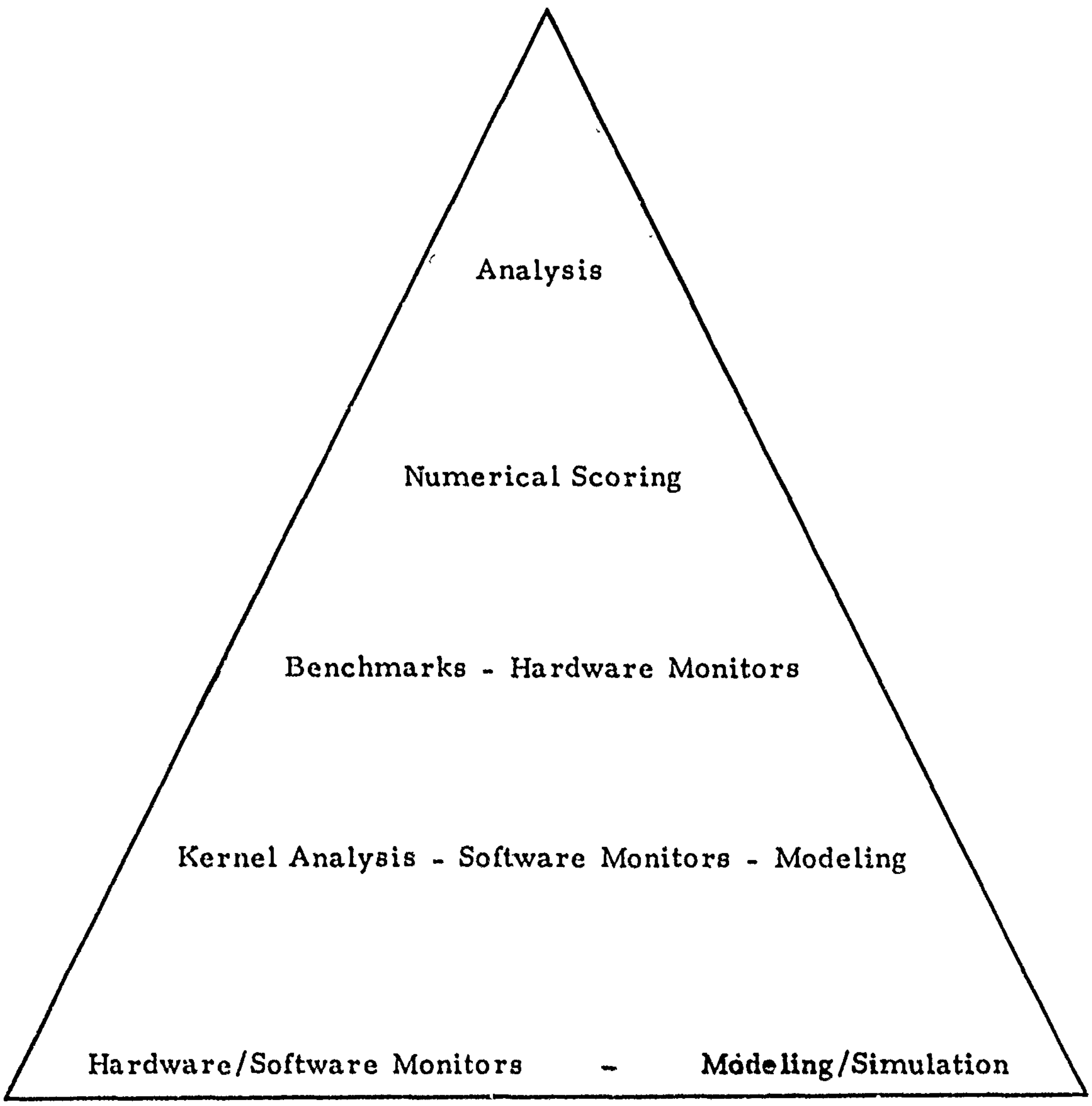

Figure 2. DMS Test Methodology Hierarchy

The hierarchial approach also permits test personnel to enter the structure at a lower level if circumstances so warrant. It is not necessary to pass through the first three levels if a user simply wishes to know which file structure would best service his requirements. Such tests can be performed by entering the hierarchy on the fourth level and utilizing one of the listed techniques; for example, the FORMS modeling technique.

Ease of entrance also means ease of egression. As soon as the test personnel are satisfied as to which DMS best satisfies their requirements, they can exit the hierarchy and, thus, terminate the test process.

(1) Step 4

The next step in the DMS test phase will be the utilization of an analysis technique to; 1) determine what DMSs are available and should be considered and 2) initially pass these systems against the requirements generated in the preparatory steps.

A most important consideration that cannot be overlooked is insuring that those responsible for the test are fully aware of all the DMSs presently available or soon to be available. It is obvious that a valid test cannot be conducted if some qualified candidates are overlooked. Therefore, some time must be expended to insure that all systems are initially considered.

These candidates then must be considered in light of the mandatory requirements generated in the first three steps. If the hardware configuration already exists or has already been selected, then some systems can be eliminated immediately. The analysis can then proceed to consider some of the more basic requirements such as file generation, maintenance, retrieval and/or output. Other systems may also fall out of the running because of some basic lack regarding these general capabilities. For example, ADVISOR, a DMS implemented on Honeywell 6000 line machines, has no file generation capability. This type of analysis presumes that a review of available documentation has already been performed. The documentation should be used only to establish the presence or lack of general capabilities and not the quality of the capability. This caution stems from the gross amount of misleading documentation distributed.

Operational systems, also, should be viewed in an actual processing environment to test information derived from the documentation. Operators, programmers and analysts functioning in the processing environment should be interviewed and their opinions noted. If possible, the system should be seen in more than one processing environment to compensate for localized faults due to a poor operating system or poor applications programs that might be observed at the first site visited.

All data gathered from this exercise would be classified as a soft measurement and, therefore, should be so treated. This data should not be used as a basis for a final decision unless one system is so outstanding that there is not doubt as to its superior qualifications. The methodology could also end after this phase if only one candidate remains. Further testing would then be worthless.

Normally, however, the analysis will result in a scaling of the candidates DMSs from the best to the worst. If no clear ranking exists from best to worst, or if a large number of systems are grouped at the top, then perhaps the utilization of another passive technique such as numerical scoring should be considered to filter out more of the less qualified systems. The reason behind this is to eliminate the need to actively test an excessive number of systems which could be quite costly in terms of time and money, not to mention the difficulty associated with implementing such a test program. The use of numerical scoring also should be considered if, for one reason or another, the facilities do not exist to actively test the DMSs. However, if after employing the analysis technique only a couple of systems are still under consideration, then the test personnel can bypass the numerical scoring technique and immediately perform some active measurement. This decision should be based on the degree to which they deemed the analysis accurate. The matrix presented in Section IV can be used again to identify the type of techniques that can be employed in subsequent testing and to update the current test plan.

(2) Step 5

Numerical scoring as exemplified by PEGS (3) is a passive techn ique in which various DMS attributes are analyzed and assigned a numerical rating based on the degree to which they satisfy the DMS requirements developed in Step 2. This is a much more structured technique than a simple analysis and is quite valuable when widely diverse applications are part of the operational environment. By rating in degree of importance the DMS parameters, the technique quantitatively forces the test personnel to rank their applications. Rating the degree to which these parameters are satisfied by a particular DMS results in a series of individual scores (by parameter) and overall scores (by system) that are easy for test personnel to interpret and apply.

Care, however, must be taken so as not to assign too much weight to the resulting scores. Because of the subjectivity involved, active testing should be performed on the top rated systems, unless one system decisively rises to the top in the rating during the test process.

The next series of techniques actively measure the performance of the DMS. They should be used; 1) to more precisely measure systems which have passed the passive stage of testing, 2) to conduct acceptance tests on an already-selected system, and 3) to identify and correct problems in already-operational systems. Regarding the latter situation, the preceding steps usually can be omitted because they involve the selection of a DMS which, in this case, has already been done. The requirements phase of the methodology may be repeated because of the possibility of a significant change in the requirements since the implementation of the DMS, but this, normally, would concern modifications to the utilization of the present system (different file structure, access methods, etc.), rather than a possible replacement of the present system with another.

(3) Step 6

The sixth step in the methodology consists of obtaining general performance data on the total software configuration which would include the operating system, the DMS and the applications programs. Benchmark programs and/or hardware monitors would be used to obtain this data, which would be used to evaluate the speed and performance of the computing system as a whole, including the DMS. Because of the presence of two variables, the operating system and the applications programs, the derived data cannot be assumed to be an absolute reflection on the quality of the DMS. In host DMSs, the problem associated with the DMS-OS interface is limited to the OS version under which the system is operating and the control of this variable is greater. For example, IDS only operates under GECOS and by simply testing the system under one of the most current versions, representative timing and performance data can be obtained, whereas DMSs that operate under a variety of operating systems make it difficult if not impossible to obtain representative performance data.

The standardization of compilers will alleviate the problem resulting from the utilization of different sets of benchmark programs to test DMS that reside in diverse machine environments. When the language dialects of compilers are standardized, then a single set of benchmark programs can be written and subsequently executed within all the different hardware/operating system environments which house the candidate DMSs.

The criteria to be employed in technique selection should be based on the problem at hand and the available facilities. For instance, benchmarks normally would be chosen for a new system evaluation because the use of a hardware monitor presumes that some sort of benchmark or actual applications programs already exist which can then be run on the monitored system. Since benchmark programs would have to be written anyway, the additional expense incurred in renting a hardware monitor could be avoided.

However, if test personnel are attempting to identify a problem area within an already-operational DMS, technique selection should be governed by the cost and ease of implementation. For example, if a hardware monitor and personnel trained in its use are available and inexpensive, then it could serve as the test vehicle If, however, it was determined that the cost of writing and executing benchmark programs was less than utilizing hardware monitors in terms of man-hours expended and the cost of computer time, then benchmarks should be employed.

If the results from this level of testing indicate that one DMS is far superior to another then the testing process can be terminated. If, however, the results are inconclusive or further information regarding the utilization of different techniques within a DMS is desired, then the testing should continue.

(4) Step 7

The seventh step in the methodology is used to further test the proficiency of one DMS vis-a-vis another in terms of their common attributes and also to evaluate the performance of various techniques within one DMS. The former case would be used for DMS selection while the latter would be used for DMS optimization.

If the results of level six testing did not clearly indicate a superior DMS, then the testing becomes more specific and those attributes common to the candidate systems that are considered the most vital vis-a-vis user requirements are measured using software monitors and/or kernel analysis. At this stage of the testing process there should be no more than a couple of candidates remaining; therefore, the use of such techniques would be practical.

Software monitors would be the recommended approach because of the flexibility provided by their utilization. Any and all parts of the DMSs can be measured with the only qualification being the quality of the documentation available. Without good documentation, the utilization of this technique is difficult if not impossible.

Kernel analysis, like software monitoring, possesses a good deal of flexibility, but the acquisition of useful timing data presumes the existence of some sort of system/user monitor or accounting system to isolate the functions to be measured and to collect timing data on them. Unless the system already performs this function, testing personnel would be required to generate some software monitors.

The utilization of such techniques to further refine the measurement data already collected might seem like an esoteric exercise, but in many processing environments, time costs money and a difference of micro-seconds between the execution of one DMS function vis-a-vis another can represent a significant cost saving or expenditure when you consider the number of times a particular module may be executed during a day. Also because of the variables involved in benchmark testing, the system seemingly with the best performance might, in fact, not be the most efficient system. Therefore, software monitors can substantiate or refute a previously arrived at decision.

Of course, one must weigh the cost involved in embedding software monitors within a couple of DMSs against the cost that might eventually be incurred if the most efficient DMS is not selected. The testing may clearly point to the best candidate before this level of testing is ever reached and it would be fruitless to continue, but if this is not the case then the decision stated above must be made.

This level of testing also could be used to optimize particular capabilities within a DMS. Software monitors, kernel analysis and modeling can be used to identify problem areas within a particular DMS and also to optimize the efficiency of the DMS. If the problem area has been loosely identified, then, by using the above mentioned techniques, the cause

of the problem can be pinpointed and perhaps corrected by experimenting with other procedures and obtaining performance data on them. For example, a software monitor may indicate that the access method that is presently being utilized is the cause of the bottleneck. Then by using a modeling technique such as FORMS, various other file structures and access methods can be simulated and performance data collected. Decisions can then be based on hard quantifiable data.

Because of the limited capabilities of the presently available modeling packages, software monitors again would be the most feasible of the three techniques, the only requirement being that good documentation of the DMS must be available.

The information gathered during the implementation of these techniques should be sufficient to reach a decision. However, if even a more thorough understanding of the system is desired and it is worth the price, then the testing could proceed to Step 8.

(5) Step 8

Step 8 consists of either using hardware monitors in conjunction with software monitors or employing a sophisticated modeling and simulation packages to derive a more complete understanding of total system performance.

These techniques are neither easy nor inexpensive to employ, but certain situations might require their utilization.

No small amount of planning is required to properly employ a combination of hardware/software monitors. The hardware monitors and system's clocks must be synchronized in order to reduce, collate and subsequently analyse the collected data. Duplication should be avoided and the monitors should be so placed as to capture the performance of the whole system. For example, while software monitors are embedded in the DMS, the hardware monitors can be determining channel, activity, CPU-I/O overlap and device activity. This data, after it is reduced and collated can then be used to reconstruct the operation of the system, including DMS-OS interface, DMS-applications programs interaction, etc.

Modeling or simulating an entire system would be almost prohibitively expensive and difficult unless pre-packaged software could be used. Such software, however, is at present neither flexible nor specific enough to accomplish the purpose associated with this level of testing. This technique was included, however, because of the capabilities that may eventually lie within the utilization of this technique.

Subsequent to the completion of this level of testing, the complete methodology has been traversed and a decision regarding the selection, acceptance, or optimization of a DMS should have been made.

If not, then, the methodology allows testing personnel to re-enter the cycle at any point. Perhaps the requirements need to be

re-examined - even the requirement for a DMS, or perhaps the data collected indicated a problem area or vital function that had not been previously investigated. Therefore, the test personnel must be allowed to retrace some of their steps and reaccomplish some of the testing with the new situation being considered. As soon as a DMS or a particular aspect of it, has been selected or accepted, the testing can stop.

c. Summary

The DMS Test Methodology appearing above is to serve as a guideline in the selection and/or acceptance of a DMS or the solution of a DMS problem. It is to be used as a tool to guide system test personnel through the logical processes that make up a test methodology. It does not attempt or suggest that the test process be completely constricted to the framework suggested by the methodology. Steps can be skipped and the hierarchy can be entered or exited at any point within the schema. The main aim is to help arrive at the selection, acceptance or optimization of a DMS and the tool can and should be molded to coincide with the purposes or each particular test and evaluation.

IV. DMS CHARACTERISTIC/TEST PAIRING

1. Introduction

The purpose of this section is to make a firm link between the DMS characteristics which were presented in Section II and the various test techniques which were described in Section III.

Part 2, Characteristic Aggregation, will discuss the situation in which many DMS characteristics are measured in aggregate by given testing techniques and ways of interpreting the results of these tests including how to isolate required unique characteristics by the use of multi-phase testing techniques.

Part 3, Characteristic/Test Pairs, will make the firm link between each of the characteristics listed in Section II and those testing techniques described in Section III which can be used to test the characteristics.

Part 4, Measurement/Test Pairs, will use the results of Part 3 to indicate in what way each test technique will be used for evaluating a particular DMS.

2. Characteristic Aggregation

Many of the DMS characteristics listed in Section II are strongly interrelated or bound together, and may be tested either in whole or in part. An example of this phenomenon would be in Section IV.3.3 - Software Facilities. This characteristic is further broken down into thirteen other characteristics, but they are all inter-related and bound together as one area of DMS characteristics.

The discriminatory powers of certain test methods are broad enough and can be used to test DMS characteristics in aggregate. Benchmark programs can be designed to measure a general area or a single characteristic of a DMS. The same can be said of all of the DMS test tools, especially those producing "hard" results.

The situation described in the preceding paragraphs results in the case where many DMS characteristics are measured in aggregate by given test tools. In some cases an aggregate measurement may give the desired results or, the evaluation team can interpret the results of the aggregate test to isolate the performance of a single characteristic. Multi-phase testing can also be a valuable tool to isolate a desired characteristic's test results when testing in an aggregate fashion. Various factors, such as hardware/software environment, test data, etc., can be varied with each test run to highlight the performance of a desired characteristic. This requires, however, that the prospective tester be familiar with the operations of the DMS being tested so that the results of the multi-phasing reflect the performance of the characteristic that is in question.

3. Characteristic/Test Pairs

This section will make a firm link between all of the DMS characteristics and the various techniques. This linking process is shown in matrix form with the DMS characteristics listed in Section II forming the rows, and the testing techniques, which are identified by letter codes, forming the columns. The letter codes assigned to the testing techniques are:

A - Benchmark Programs
B - Modeling
C - Simulation
D - Hardware Monitor
E - Software Monitor
F - Documentation Analysis
G - Operational Analysis
H - Numerical Scoring Method

The characteristics and test techniques will be linked together by the presence of either an "H" or "S" in the applicable column. The letter "H" indicates that the results of the test will be "Hard" and "S" indicates "Soft" test results.

Also included in the matrix is a reference number which references the matrix shown in Part 4, "Measurement/Test Pairs," which is described in the next section.

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
I. Data Structure and Definition									I
A. Logical Structure		H	H			S	S		I. I
1. Items									I. I. 1
a. System term for item								S	I. I. 1. 1
b. Item types provided									I. I. 1. 2
(1) Numeric item types									I. I. 1. 2. 1
(a) System term (packed decimal, fixed binary, right justified EBCDIC, etc.)						S		S	I. I. 1. 2. 1. 1
(b) Storage representation						S		S	I. I. 1. 2. 1. 2
(c) Length						S		S	I. I. 1. 2. 1. 3
(d) Numeric sub-items permitted								S	I. I. 1. 2. 1. 4
(2) String item types									I. I. 1. 2. 2
(a) System term (alphanumeric, left justified EBCDIC, variable length, etc.)						S		S	I. I. 1. 2. 2. 1
(b) Storage representation						S		S	I. I. 1. 2. 2. 2
(c) Length						S		S	I. I. 1. 2. 2. 3
(d) String sub-items permitted								S	I. I. 1. 2. 2. 4
(3) Other item types									I. I. 1. 2. 3
(a) Date								S	I. I. 1. 2. 3. 1
(b) Coordinates								S	I. I. 1. 2. 3. 2
(c) Boolean								S	I. I. 1. 2. 3. 3
c. Item value type attributes									I. I. 1. 3
(1) Numeric								S	I. I. 1. 3. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(a) Fixed length						S		S	I. I. 1. 3. 1. 1
(b) Length range						S		S	I. I. 1. 3. 1. 2
(c) Picture									I. I. 1. 3. 1. 3
(1 Positioning								S	I. I. 1. 3. 1. 3. 1
(2 Precision								S	I. I. 1. 3. 1. 3. 2
(3 Right/left justification								S	I. I. 1. 3. 1. 3. 3
(d) Value range								S	I. I. 1. 3. 1. 4
(e) Value list								S	I. I. 1. 3. 1. 5
(f) Validation routine								S	I. I. 1. 3. 1. 6
(2) String									I. I. 1. 3. 2
(a) Fixed length								S	I. I. 1. 3. 2. 1
(b) Length range						S		S	I. I. 1. 3. 2. 2
(c) Picture									I. I. 1. 3. 2. 3
(1 Positioning								S	I. I. 1. 3. 2. 3. 1
(2 Right/left justification								S	I. I. 1. 3. 2. 3. 2
(d) Value range								S	I. I. 1. 3. 2. 4
(e) Value list								S	I. I. 1. 3. 2. 5
(f) Validation routine								S	I. I. 1. 3. 2. 6
d. Item identification									I. I. 1. 4
(1) Length								S	I. I. 1. 4. 1
(2) Mode								S	I. I. 1. 4. 2
(3) Embedded blanks								S	I. I. 1. 4. 3
(4) Synonyms								S	I. I. 1. 4. 4
(5) Unique within:									I. I. 1. 4. 5
(a) Group								S	I. I. 1. 4. 5. 1
(b) Entry								S	I. I. 1. 4. 5. 2

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(6) Output heading								S	I. I. 1. 4. 6
e. Security									I. I. 1. 5
(1) Access locks controlling access during:									I. I. 1. 5. 1
(a) Query function								S	I. I. 1. 5. 1. 1
(b) Update function								S	I. I. 1. 5. 1. 2
(2) Program identification codes for use of item during:									I. I. 1. 5. 2
(a) Query function								S	I. I. 1. 5. 2. 1
(b) Update function								S	I. I. 1. 5. 2. 2
f. Other item attributes									I. I. 1. 6
(1) Units of expression (feet, pounds, dollars, etc.)								S	I. I. 1. 6. 1
(2) Usage of item values								S	I. I. 1. 6. 2
(3) Output editing attributes						S		S	I. I. 1. 6. 3
(4) Input and output conversion attributes						S		S	I. I. 1. 6. 4
(5) Value synonym lists								S	I. I. 1. 6. 5
g. Non-structural attributes									I. I. 1. 7
(1) Existence indicators								S	I. I. 1. 7. 1
(2) Date and time of change to item value								S	I. I. 1. 7. 2
(3) Identification of transaction or program supplying current item value								S	I. I. 1. 7. 3
(4) Null item values								S	I. I. 1. 7. 4
(5) Multiple item values								S	I. I. 1. 7. 5
2. Groups									I. I. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
a. System term for group								S	I. I. 2. 1
b. Group types:									I. I. 2. 2
(1) Simple repeating								S	I. I. 2. 2. 1
(2) Simple non-repeating								S	I. I. 2. 2. 2
(3) Compound repeating								S	I. I. 2. 2. 3
(4) Compound non-repeating								S	I. I. 2. 2. 4
c. Group identification									I. I. 2. 3
(1) Effective length								S	I. I. 2. 3. 1
(2) Mode								S	I. I. 2. 3. 2
(3) Embedded blanks								S	I. I. 2. 3. 3
(4) Synonyms								S	I. I. 2. 3. 4
(5) Use of group identifiers or sequencers:									I. I. 2. 3. 5
(a) Required or optional						S		S	I. I. 2. 3. 5. 1
(b) Number of items used						S		S	I. I. 2. 3. 5. 2
(c) Ascending and/or descending sequence						S		S	I. I. 2. 3. 5. 3
d. Security									I. I. 2. 4
(1) Access locks controlling access during									I. I. 2. 4. 1
(a) Query function								S	I. I. 2. 4. 1. 1
(b) Update function								S	I. I. 2. 4. 1. 2
(2) Program identification codes for use of group during:									I. I. 2. 4. 2
(a) Query function								S	I. I. 2. 4. 2. 1
(b) Update function								S	I. I. 2. 4. 2. 2
e. Group Composition									I. I. 2. 5

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(1) Number and type of constituent items						S		S	I. I. 2. 5. 1
(2) Number and type of consituent items and groups in compound group						S		S	I. I. 2. 5. 2
(3) Depth of nesting in compound group								S	I. I. 2. 5. 3
(4) Hierarchical structure composed:								S	I. I. 2. 5. 4
(a) Linear								S	I. I. 2. 5. 4. 1
(b) Tree								S	I. I. 2. 5. 4. 2
(c) Number of levels of subordination						S		S	I. I. 2. 5. 4. 3
(d) Number of dependent groups per parent group						S		S	I. I. 2. 5. 4. 4
(e) Number of peer groups at same level of subordination						S		S	I. I. 2. 5. 4. 5
f. Non-structural attributes									I. I. 2. 6
(1) Date and time of group insertion								S	I. I. 2. 6. 1
(2) Count of references made to group								S	I. I. 2. 6. 2
3. Group relations									I. I. 3
a. System term for group relation								S	I. I. 3. 1
b. Group relation type:									I. I. 3. 2
(1) Non-hierarchic								S	I. I. 3. 2. 1
(2) Hierarchic:									I. I. 3. 2. 2
(a) Linear								S	I. I. 3. 2. 2. 1
(b) Tree								S	I. I. 3. 2. 2. 2
c. Group relation identification									I. I. 3. 3
(1) Length								S	I. I. 3. 3. 1
(2) Mode								S	I. I. 3. 3. 2
(3) Embedded blanks								S	I. I. 3. 3. 3

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(4) Synonyms								S	I. I. 3. 3. 4
d. Security									I. I. 3. 4
(1) Access locks controlling access to parent or dependent groups during:									I. I. 3. 4. 1
(a) Query function								S	I. I. 3. 4. 1. 1
(b) Update function								S	I. I. 3. 4. 1. 2
(2) Program identification codes for use of parent or dependent groups during:									I. I. 3. 4. 2
(a) Query function								S	I. I. 3. 4. 2. 1
(b) Update function								S	I. I. 3. 4. 2. 2
e. Group relation composition									I. I. 3. 5
(1) Number of levels of subordination of composite groups						S		S	I. I. 3. 5. 1
(2) Number of dependent groups per parent group						S		S	I. I. 3. 5. 2
(3) Number of peer groups at same level of subordination						S		S	I. I. 3. 5. 3
(4) Placement criteria for insertion of new constituent group occurrence						S		S	I. I. 3. 5. 4
f. Non-structural attributes									I. I. . 6
(1) Count of constituent groups or repetitions of a constituent group								S	I. I. 3. 6. 1
4. Entries									I. I. 4
a. System term for entry								S	I. I. 4. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
b. Entry type:									I. I. 4. 2
(1) Single type entry only								S	I. I. 4. 2. 1
(2) Multiple entry types provided								S	I. I. 4. 2. 2
c. Entry identification									I. I. 4. 3
(1) Length								S	I. I. 4. 3. 1
(2) Mode								S	I. I. 4. 3. 2
(3) Embedded blanks permitted								S	I. I. 4. 3. 3
(4) Synonyms permitted								S	I. I. 4. 3. 4
(5) Item sets serving as identifier or sequencer								S	I. I. 4. 3. 5
d. Security									I. I. 4. 4
(1) Access locks for controlling access to entry instances during:									I. I. 4. 4. 1
(a) Query function								S	I. I. 4. 4. 1. 1
(b) Update function								S	I. I. 4. 4. 1. 2
(2) Codes which identify programs using the entry during:									I. I. 4. 4. 2
(a) Query function								S	I. I. 4. 4. 2. 1
(b) Update function								S	I. I. 4. 4. 2. 2
e. Entry composition									I. I. 4. 5
(1) Limitation on number and type of group and group relationships comprising the entry						S		S	I. I. 4. 5. 1
(2) Limitation on the number of hierarchic levels in the entry						S		S	I. I. 4. 5. 2
(3) Limitation in number of relations in which a dependent group may participate in the entry.						S		S	I. I. 4. 5. 3

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
f. Non-structural attributes									I. I. 4. 6
(1) Date and time entry was placed in the file . . .								S	I. I. 4. 6. 1
(2) Count of references to entry								S	I. I. 4. 6. 2
5. Files									I. I. 5
a. System term for file								S	I. I. 5. 1
b. File types:									I. I. 5. 2
(1) Single file type only								S	I. I. 5. 2. 1
(2) Multiple file types provided:									I. I. 5. 2. 2
(a) Differentiated by function								S	I. I. 5. 2. 2. 1
(b) Differentiated by sets of attributes								S	I. I. 5. 2. 2. 2
c. File identification									I. I. 5. 3
(1) Length								S	I. I. 5. 3. 1
(2) Mode								S	I. I. 5. 3. 2
(3) Embedded blanks								S	I. I. 5. 3. 3
(4) Synonyms:								S	I. I. 5. 3. 4
(a) Maximum number of synonyms						S		S	I. I. 5. 3. 4. 1
(b) File can be accessed by synonym (not restricted to output)								S	I. I. 5. 3. 4. 2
d. Security									I. I. 5. 4
(1) Access locks for controlling access to file during:									I. I. 5. 4. 1
(a) Update function								S	I. I. 5. 4. 1. 1
(b) Query function								S	I. I. 5. 4. 1. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
e. File composition									I. I. 5. 5
(1) Number of files in data base						S		S	I. I. 5. 5. 1
(2) Number and type of entries in file						S		S	I. I. 5. 5. 2
(3) Limitations on inter-entry relations						S		S	I. I. 5. 5. 3
f. Non-structured attributes									I. I. 5. 6
(1) File instance numbers								S	I. I. 5. 6. 1
(2) Date-time stamp								S	I. I. 5. 6. 2
(3) Entry counts								S	I. I. 5. 6. 3
(4) Control totals over file entries								S	I. I. 5. 6. 4
E. Data Definition									I. II
1. Data Definition Language									I. II. 1
a. Form used (narrative, keyword, fixed position, separator)						S		S	I. II. 1. 1
b. Form same as language used by other DMS functions								S	I. II. 1. 2
c. Same language is used to define all data elements								S	I. II. 1. 3
2. Context of Data Definition									I. II. 2
a. Function is an integral part of each program processing the data and is compiled with it								S	I. II. 2. 1
b. Function is integrated with file creation function								S	I. II. 2. 2
c. Function is input to system as a separate function								S	I. II. 2. 3
d. Lines of data definition can be input in any order								S	I. II. 2. 4
3. Data Definition Revision	H				H		S		I. II. 3
a. Re-entrance of entire definition required	H							S	I. II. 3. 1
b. Input of changes only required	H							S	I. II. 3. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
c. Data values are moved or changed during revision	H							S	I. II. 3. 3
d. Restructuring of stored data must be provided by user	H							S	I. II. 3. 4
e. System provides automatic restructuring of stored data	H							S	I. II. 3. 5
f. Existing definition may be deleted for which attributes of:									I. II. 3. 6
(1) Item	H					S		S	I. II. 3. 6. 1
(2) Group	H					S		S	I. II. 3. 6. 2
(3) Group relation	H					S		S	I. II. 3. 6. 3
(4) Entry	H					S		S	I. II. 3. 6. 4
(5) File	H					S		S	I. II. 3. 6. 5
g. Existing definition may be expanded or modified for which of the above (I-B-3-f-1 through 5) data elements	H					S		S	I. II. 3. 7
h. Entire definition may be deleted or replaced for which of the above (I-B-3-f-1 through 5) data elements	H					S		S	I. II. 3. 8
4. Definition of Components	H				H		S		I. II. 4
a. Item definition	H								I. II. 4. 1
(1) Identification									I. II. 4. 1. 1
(a) Item name								S	I. II. 4. 1. 1. 1
(b) Item number								S	I. II. 4. 1. 1. 2
(c) Synonym								S	I. II. 4. 1. 1. 3
(2) Security									I. II. 4. 1. 2
(a) Requirements for user access during:									I. II. 4. 1. 2. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(1 Update function								S	I. II. 4. 1. 2. 1. 1
(2 Query function								S	I. II. 4. 1. 2. 1. 2
b. Item value definition	H				H		S		I. II. 4. 2
(1) Item value type								S	I. II. 4. 2. 1
(2) Item value length:									I. II. 4. 2. 2
(a) Specified in digits, bytes, characters								S	I. II. 4. 2. 2. 1
(b) Fixed in system								S	I. II. 4. 2. 2. 2
(3) Item value placement (left or right justification)									I. II. 4. 2. 3
(a) Starting character position of item value								S	I. II. 4. 2. 3. 1
(b) Definition of value range:									I. II. 4. 2. 3. 2
(1 Minimum/maximum value								S	I. II. 4. 2. 3. 2. 1
(2 Item value list								S	I. II. 4. 2. 3. 2. 2
(c) Attributes defined in picture statement								S	I. II. 4. 2. 3. 3
(d) Editing directions for input/output or for validation								S	I. II. 4. 2. 3. 4
(1 Defined in picture statement								S	I. II. 4. 2. 3. 4. 1
(2 Specified by separate routines or tables								S	I. II. 4. 2. 3. 4. 2
(3 Capabilities include:									I. II. 4. 2. 3. 4. 3
-a- Automatic truncation								S	I. II. 4. 2. 3. 4. 3. 1
-b- Automatic padding								S	I. II. 4. 2. 3. 4. 3. 2
-c- Additional information added to data								S	I. II. 4. 2. 3. 4. 3. 3
-d- Encoding of values								S	I. II. 4. 2. 3. 4. 3. 4
-e- Left justification of field								S	I. II. 4. 2. 3. 4. 3. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
-f- Zero suppression								S	I. II. 4. 2. 3. 4. 3. 6
-g- Truncation								S	I. II. 4. 2. 3. 4. 3. 7
-h- Algebraic sign								S	I. II. 4. 2. 3. 4. 3. 8
-i- Punctuation								S	I. II. 4. 2. 3. 4. 3. 9
-j- Percent								S	I. II. 4. 2. 3. 4. 3. 10
-k- Engineering notation								S	I. II. 4. 2. 3. 4. 3. 11
-l- Dollar sign								S	I. II. 4. 2. 3. 4. 3. 12
-m- Specific characters								S	I. II. 4. 2. 3. 4. 3. 13
-n- Decoding of values								S	I. II. 4. 2. 3. 4. 3. 14
-o- Data conversions								S	I. II. 4. 2. 3. 4. 3. 15
-p- Standard measurement conversions.								S	I. II. 4. 2. 3. 4. 3. 16
-q- Scaling								S	I. II. 4. 2. 3. 4. 3. 17
-r- Rounding								S	I. II. 4. 2. 3. 4. 3. 18
-s- Item size modification								S	I. II. 4. 2. 3. 4. 3. 19
-t- Right justification of field . .								S	I. II. 4. 2. 3. 4. 3. 20
(4) Item value definition is specified at same time as item definition								S	I. II. 4. 2. 4
(5) Item value definition is in separate section of the data definition								S	I. II. 4. 2. 5
c. Subitem definition	H				H		S		I. II. 4. 3
(1) Identification								S	I. II. 4. 3. 1
(2) Length.								S	I. II. 4. 3. 2
(3) Starting position								S	I. II. 4. 3. 3
d. Group definition	H				H		S		I. II. 4. 4
(1) Item and group attributes defined separately for:									I. II. 4. 4. 1
(a) Repeating group								S	I. II. 4. 4. 1. 1
(b) Non-repeating group								S	I. II. 4. 4. 1. 2

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(2) Relationship of constituent item/group definitions to parent group:									I. II. 4. 4. 2
(a) All have same level or group number . .								S	I. II. 4. 4. 2. 1
(b) Definition ordering (group is defined to consist of all items whose definitions immediately follow or precede the group definition statement)								S	I. II. 4. 4. 2. 2
(c) Top-down or bottom-up approach								S	I. II. 4. 4. 2. 3
(3) Identification.								S	I. II. 4. 4. 3
(4) Security:									I. II. 4. 4. 4
(a) Requirements for user access during:. .									I. II. 4. 4. 4. 1
(1 Update function								S	I. II. 4. 4. 4. 1. 1
(2 Query function								S	I. II. 4. 4. 4. 1. 2
(5) Group sequence item:									I. II. 4. 4. 5
(a) Number of items in sequencer set						S		S	I. II. 4. 4. 5. 1
(b) Ascending or descending sequence may be specified								S	I. II. 4. 4. 5. 2
(6) Explicit group level number:								S	I. II. 4. 4. 6
(a) Group and items or groups only								S	I. II. 4. 4. 6. 1
(7) Count item								S	I. II. 4. 4. 7
(8) Special keyword for group:								S	I. II. 4. 4. 8
(a) Repeating group								S	I. II. 4. 4. 8. 1
(b) Non-repeating group								S	I. II. 4. 4. 8. 2
e. Group relation definition	H		H			S			I. II. 4. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(1) Identification								S	I. II. 4. 5. 1
(2) Security									I. II. 4. 5. 2
(a) Requirements for user access during:									I. II. 4. 5. 2. 1
(1 Update function								S	I. II. 4. 5. 2. 1. 1
(2 Query function								S	I. II. 4. 5. 2. 1. 2
(3) Definition is implicit (it is implied by the definition of the hierarchical entry structure).								S	I. II. 4. 5. 3
(4) Definition is explicit (ordering is stated in definition statement)								S	I. II. 4. 5. 4
(5) Definition is separate from group definition								S	I. II. 4. 5. 5
(6) Definition is part of group definition								S	I. II. 4. 5. 6
f. Entry definition	H				H		S		I. II. 4. 6
(1) Identification								S	I. II. 4. 6. 1
(2) Security									I. II. 4. 6. 2
(a) Requirements for user access during:									I. II. 4. 6. 2. 1
(1 Update function								S	I. II. 4. 6. 2. 1. 1
(2 Query function								S	I. II. 4. 6. 2. 1. 2
(3) Organization of constituent groups:									I. II. 4. 6. 3
(a) Specified in group or group relation definition								S	I. II. 4. 6. 3. 1
(b) Specified separately in entry definition								S	I. II. 4. 6. 3. 2
(4) Multiple logical data structures can be defined within one entry								S	I. II. 4. 6. 4
g. File definition	H				H		S		I. II. 4. 7

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(1) Identification								S	I. II. 4. 7. 1
(2) Security									I. II. 4. 7. 2
(a) Requirements for user access during:									I. II. 4. 7. 2. 1
(1 Update function								S	I. II. 4. 7. 2. 1. 1
(2 Query function								S	I. II. 4. 7. 2. 1. 2
(3) Sequence of constituent definitions						S		S	I. II. 4. 7. 3
C. Storage Structure	H	H	H	H	H	S	S		I. III
1. Control									I. III. 1
a. Host environment dependencies									I. III. 1. 1
(1) File storage device is automatically assigned if no special request is made								S	I. III. 1. 1. 1
(2) DMS uses operating system supplied access methods								S	I. III. 1. 1. 2
(3) DMS uses operating system supplied space and resource management (allocation of buffers, provision of additional storage in the event of overflow)								S	I. III. 1. 1. 3
(4) Operating system provides for storage device interchangeability								S	I. III. 1. 1. 4
b. User control of storage structure									I. III. 1. 2
(1) Devices available (see HOST ENVIRONMENT INTERRELATIONSHIPS):									I. III. 1. 2. 1
(a) Sequential devices		H	H			S		S	I. III. 1. 2. 1. 1
(b) Direct access devices		H	H			S		S	I. III. 1. 2. 1. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(2) Sequential or direct access can be chosen: . .								S	I. III. 1. 2. 2
(a) Operating system supplied access methods can be chosen	H	H	H	H	H		S	S	I. III. 1. 2. 2. 1
(b) DMS augments O/S supplied access methods	H	H	H	H	H		S	S	I. III. 1. 2. 2. 2
(3) User control over space allocation for file on storage medium									I. III. 1. 2. 3
(a) Length								S	I. III. 1. 2. 3. 1
(b) Variable or fixed								S	I. III. 1. 2. 3. 2
(c) Blocking factor								S	I. III. 1. 2. 3. 3
(d) Maximum and average size								S	I. III. 1. 2. 3. 4
(e) Number of entries per record								S	I. III. 1. 2. 3. 5
(f) Entry length								S	I. III. 1. 2. 3. 6
(g) Paging technique (storage subdivided into specified number of characters)								S	I. III. 1. 2. 3. 7
(4) User control over index arrangement						S		S	I. III. 1. 2. 4
(5) User supplied random accessing formula is necessary								S	I. III. 1. 2. 5
2. Storage Structure Representation									I. III. 2
a. Logical sequential (each group instance is stored according to its subordinate relationship starting with the master group)	H	H	H		H		S	S	I. III. 2. 1
b. Sequential (elements of data follow one after the other but do not conform to the logical organization of the file):	H	H	H		H		S	S	I. III. 2. 2
(1) All values of an item are stored at any vacant place in the file, saving the address of that location for further use in retrieval) . .	H	H	H		H		S	S	I. III. 2. 2. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
3. Organization techniques									I. III. 3
a. DMS augments O/S supplied access methods to provide better indexing for hierarchical files	H	H	H		H		S	S	I. III. 3. 1
b. Techniques for establishing logical sequential organization include:									I. III. 3. 2
(1) Table index (an ordered reference list to the contents of a file)								S	I. III. 3. 2. 1
(2) Chaining:					H		S	S	I. III. 3. 2. 2
(a) Groups are chained together using top-down method (chained to next group at same level)	H				H			S	I. III. 3. 2. 2. 1
(b) Groups are chained together using top-down method (dependent groups follow superior groups; the last group in one string is chained to the first group of the next higher level string)	H				H			S	I. III. 3. 2. 2. 2
(c) Chaining is combined with table indexing	H				H			S	I. III. 3. 2. 2. 3
(d) Pointers are comprised of:									I. III. 3. 2. 2. 4
(1) Address of a physical record	H				H			S	I. III. 3. 2. 2. 4. 1
(2) Address of a group	H				H			S	I. III. 3. 2. 2. 4. 2
(3) Item value	H				H			S	I. III. 3. 2. 2. 4. 3
(e) Multiple pointers are used		H	H		H			S	I. III. 3. 2. 2. 5
(f) Storage allocation for pointers								S	I. III. 3. 2. 2. 6
4. Storage Access Methods	H	H	H	H	H	S	S		I. III. 4
a. Indexing (the association of a data value with the address of an element containing that value)	H	H	H	H	H		S	S	I. III. 4. 1

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(1) Performed through operating system access methods								S	I. III. 4. 1. 1
(a) O/S provides overflow for index entries not fitting into memory								S	I. III. 4. 1. 1. 1
(2) DMS provided indexing features:	H								I. III. 4. 1. 2
(a) Index contains one value-address pair for every occurrence of a given value					H			S	I. III. 4. 1. 2. 1
(b) Index references only ranges of values					H			S	I. III. 4. 1. 2. 2
(c) Inverted file used -- file indexed by every value of every item					H			S	I. III. 4. 1. 2. 3
b. Randomizing or hash coding -- the transformation of item values in the group or entry into an address	H	H	H	H	H		S	S	I. III. 4. 2
(1) User supplied randomizing formula necessary or may be specified to replace O/S supplied formula	H	H	H		H			S	I. III. 4. 2. 1
(2) Formula can handle synonyms (records whose control field happens to randomize to the same address as other records)		H	H		H			S	I. III. 4. 2. 2
(a) Synonyms are placed in overflow area provided		H	H		H			S	I. III. 4. 2. 2. 1
(3) Formula is affected by file size	H	H	H		H			S	I. III. 4. 2. 3
(4) Formula is affected by volatility or stability of data base	H	H	H		H			S	I. III. 4. 2. 4
II. Data Manipulation Functions	H	H	H	H	H	S	S		II
A. File Creation	H	H	H						II. I

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
1. Input file definition									II. I. 1
a. System specified format required								S	II. I. 1. 1
b. User specified format permitted								S	II. I. 1. 2
c. Same facilities provided for definition of master file								S	II. I. 1. 3
d. Same facilities provided by update function for defining transactions								S	II. I. 1. 4
e. User can specify access method for input file								S	II. I. 1. 5
f. Data and storage structure definition performed in single task								S	II. I. 1. 6
2. Allocation of media space	H	H	H						II. I. 2
a. Operating system facilities provide for:	H	H	H						II. I. 2. 1
(1) Space allocation for input file								S	II. I. 2. 1. 1
(2) Space allocation for master file								S	II. I. 2. 1. 2
(3) Interchangeability of device type								S	II. I. 2. 1. 3
(4) Overflow areas								S	II. I. 2. 1. 4
(5) Diagnostics						S		S	II. I. 2. 1. 5
b. Utility programs provided								S	II. I. 2. 2
c. User specifications required:									II. I. 2. 3
(1) Device type								S	II. I. 2. 3. 1
(2) Blocking method								S	II. I. 2. 3. 2
(3) Space reservation for:									II. I. 2. 3. 3
(a) File size								S	I. I. 2. 3. 3. 1
(b) Overflow area								S	II. I. 2. 3. 3. 2
(c) Working area								S	II. I. 2. 3. 3. 3
3. Provision of input data file								S	II. I. 3

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
a. Acceptability of input date files generated on other computers or under different operating system: . .									II. I. 3. 1
(1) Foreign files accepted								S	II. I. 3. 1. 1
(2) Physical media:									II. I. 3. 1. 2
(a) Magnetic tape								S	II. I. 3. 1. 2. 1
(b) Disc								S	II. I. 3. 1. 2. 2
(c) Other						S		S	II. I. 3. 1. 2. 3
b. Acceptability of files produced by other system processors within the operating system under which the DMS operates:									II. I. 3. 2
(1) COBOL							S	S	II. I. 3. 2. 1
(2) FORTRAN							S	S	II. I. 3. 2. 2
(3) PL/1							S	S	II. I. 3. 2. 3
(4) ALGOL							S	S	II. I. 3. 2. 4
(5) Other						S	S	S	II. I. 3. 2. 5
c. Acceptability of input data constructed from existing system files by the use of the interrogation function							S	S	II. I. 3. 3
d. Acceptability of input data as a stream of transactions							S	S	II. I. 3. 4
e. Hardware environment									II. I. 3. 5
(1) DMS can accept input data from the following local sequential devices:									II. I. 3. 5. 1
(a) Card reader								S	II. I. 3. 5. 1. 1
(b) Paper tape								S	II. I. 3. 5. 1. 2
(c) Magnetic tape								S	II. I. 3. 5. 1. 3

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(2) DMS can accept input data from local keyboard devices								S	II. I. 3. 5. 2
(3) DMS can accept input data from the following remote sequential devices:									II. I. 3. 5. 3
(a) Card reader								S	II. I. 3. 5. 3. 1
(b) Paper tape								S	II. I. 3. 5. 3. 2
(c) Magnetic tape								S	II. I. 3. 5. 3. 3
(4) DMS can accept input data from the following local direct access devices:									II. I. 3. 5. 4
(a) Drum								S	II. I. 3. 5. 4. 1
(b) Disc								S	II. I. 3. 5. 4. 2
(c) Data cell								S	II. I. 3. 5. 4. 3
(5) DMS can accept and incorporate format descriptions for input data from:									II. I. 3. 5. 5
(a) Card reader								S	II. I. 3. 5. 5. 1
(b) Paper tape								S	II. I. 3. 5. 5. 2
(c) Magnetic tape								S	II. I. 3. 5. 5. 3
(d) Keyboard device								S	II. I. 3. 5. 5. 4
(e) Drum								S	II. I. 3. 5. 5. 5
(f) Disc								S	II. I. 3. 5. 5. 6
(g) Data cell								S	II. I. 3. 5. 5. 7
(6) DMS can accept and incorporate file descriptions from:									II. I. 3. 5. 6
(a) Card reader								S	II. I. 3. 5. 6. 1
(b) Paper tape								S	II. I. 3. 5. 6. 2
(c) Magnetic tape								S	II. I. 3. 5. 6. 3
(d) Keyboard device								S	II. I. 3. 5. 6. 4
(e) Drum								S	II. I. 3. 5. 6. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(f) Disc								S	II. I. 3. 5. 6. 6
(g) Data cell								S	II. I. 3. 5. 6. 7
(7) DMS can accept and incorporate file descriptions from a remote:									II. I. 3. 5. 7
(a) Card reader								S	II. I. 3. 5. 7. 1
(b) Paper tape								S	II. I. 3. 5. 7. 2
(c) Magnetic tape								S	II. I. 3. 5. 7. 3
(d) Keyboard device								S	II. I. 3. 5. 7. 4
f. Multi-file input capabilities						S		S	II. I. 3. 6
g. Format requirements									II. I. 3. 7
(1) Direct correspondence between the sequence of the input data and the logical organization of the file to be generated								S	II. I. 3. 7. 1
(2) Several different formats for the input data during file creation are permitted								S	II. I. 3. 7. 2
(3) If data is in machine readable form, system can review it to determine transformation, restructuring or editing requirements								S	II. I. 3. 7. 3
(4) File may be prepared through use of interrogation function								S	II. I. 3. 7. 4
4. File population	H	H	H		H		S		II. I. 4
a. Accomplished through update function								S	II. I. 4. 1
b. Accomplished through separate creation functions								S	II. I. 4. 2
c. Validation facilities provided:								S	II. I. 4. 3
(1) Item value validation criteria (see Data Definition)								S	II. I. 4. 3. 1
(2) Sequence check of entries								S	II. I. 4. 3. 2
(3) Size check of entries								S	II. I. 4. 3. 3

(The printed header spans columns A–H as "TECHNIQUE".)

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
d. Data transformation:									II. I. 4. 4
(1) Transformation of item values								S	II. I. 4. 4. 1
(2) Encoding, decoding of item values								S	II. I. 4. 4. 2
(3) Reordering of items, group, entries								S	II. I. 4. 4. 3
e. Monitoring and error detection facilities provided during file population						S		S	II. I. 4. 5
5. Creation function monitoring		H	H		H				II. I. 5
a. System detected errors								S	II. I. 5. 1
b. Operating system detected errors								S	II. I. 5. 2
c. User specified conditions for error detection								S	II. I. 5. 3
d. On-line corrections permitted								S	II. I. 5. 4
e. Batched corrections permitted								S	II. I. 5. 5
f. System statistics reports								S	II. I. 5. 6
g. Restart and recovery procedures provided								S	II. I. 5. 7
h. Run history of recognized errors provided								S	II. I. 5. 8
B. File Update	H	H	H	H	H		S		II. II
1. More than one update mode exists within the system characterized by differing:									II. II. 1
a. Functions								S	II. II. 1. 1
b. Languages								S	II. II. 1. 2
c. Input media								S	II. II. 1. 3
d. Amount of user control								S	II. II. 1. 4
2. Sources									II. II. 2
a. Input medium for transaction(see FILE CREATION, Provision of Input Data File)								S	II. II. 2. 1
b. Batch processing may use:									II. II. 2. 2

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(1) Prestored infor ion:									II. II. 2. 2. 1
(a) ansactions								S	II. II. 2. 2. 1. 1
(b) Transaction definitions								S	II. II. 2. 2. 1. 2
(c) Transaction programs								S	II. II. 2. 2. 1. 3
(d) File update procedures may be prestored								S	II. II. 2. 2. 1. 4
(1 In source form								S	II. II. 2. 2. 1. 4. 1
(2 In object form (system library)								S	II. II. 2. 2. 1. 4. 2
(e) Prestored update procedures may be modified either permanently or temporarily								S	II. II. 2. 2. 1. 5
(1 Stored library form (the procedure as stored on the library is modified and the change is permanent)								S	II. II. 2. 2. 1. 5. 1
(2 Temporary modification can be made at run time and is effective only for the current run								S	II. II. 2. 2. 1. 5. 2
(f) Parameterized procedures can be prestored and the parameters entered at execution time								S	II. II. 2. 2. 1. 6
(2) At the beginning of the transaction stream:									II. II. 2. 2. 2
(a) Transaction definitions								S	II. II. 2. 2. 2. 1
(b) Transaction programs								S	II. II. 2. 2. 2. 2
(3) In the transactions:									II. II. 2. 2. 3
(a) Transaction definitions								S	II. II. 2. 2. 3. 1
(b) Transaction programs								S	II. II. 2. 2. 3. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
c. Remote terminal processing may use:									II. II. 2. 3
(1) (Same as II-B-2-b above)								S	II. II. 2. 3. 1
d. Transaction ordering									II. II. 2. 4
(1) User responsibility required by the system:								S	II. II. 2. 4. 1
(a) For all processing modes								S	II. II. 2. 4. 1. 1
(2) Transactions can be processed in any order								S	II. II. 2. 4. 2
(3) System performs presort	H							S	II. II. 2. 4. 3
3. Transaction definition									II. II. 3
a. Facilities described under Data Definitions available to transaction definition								S	II. II. 3. 1
b. Format and placement									II. II. 3. 2
(1) Format(s) used (fixed, tabular, narrative, etc.)						S		S	II. II. 3. 2. 1
(2) System required format								S	II. II. 3. 2. 2
(3) Separation of transaction and transaction definition required								S	II. II. 3. 2. 3
(4) Format and placement requirements differ according to:									II. II. 3. 2. 4
(a) Update mode used								S	II. II. 3. 2. 4. 1
(b) Transaction data element defined								S	II. II. 3. 2. 4. 2
(5) Data mapping is a function of transaction definition								S	II. II. 3. 2. 5
(6) Data validation features provided						S		S	II. II. 3. 2. 6
(7) Data editing and transformation features provided						S		S	II. II. 3. 2. 7

ATTRIBUTES	TECHNIQUE								REFERENCE NO
	A	B	C	D	E	F	G	H	
4. Transaction program	H	H	H						II. II. 4
a. Facilities described under Language Attributes of Self-Contained Systems available to the Transaction program								S	II. II. 4. 1
b. Format and placement:									II. II. 4. 2
(1) Formats used						S		S	II. II. 4. 2. 1
(2) System required format								S	II. II. 4. 2. 2
(3) Separation of transaction program and transaction definition								S	II. II. 4. 2. 3
(4) Format and placement requirements differ according to update mode used								S	II. II. 4. 2. 4
c. Data mapping is a function of the transaction program								S	II. II. 4. 3
d. Data validation features provided						S		S	II. II. 4. 4
e. Data editing and transformation features provided						S		S	II. II. 4. 5
5. Data access and manipulation	H	H	H	H	H		S		II. II. 5
a. Data access	H	H	H	H	H		S		II. II. 5. 1
(1) Automatic capabilities (no user specification)									II. II. 5. 1. 1
(a) Automatic transaction and entry access by matching identifiers								S	II. II. 5. 1. 1. 1
(b) Automatic file reading				H	H			S	II. II. 5. 1. 1. 2
(c) Automatic file writing				H	H			S	II. II. 5. 1. 1. 3
(d) Automatic capabilities differ according to update mode used								S	II. II. 5. 1. 1. 4
(2) User control									II. II. 5. 1. 2
(a) File reading:				H					II. II. 5. 1. 2. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(1 User control requirements differ according to update mode used . . .								S	II. II. 5. 1. 2. 1. 1
(b) File writing:				H					II. II. 5. 1. 2. 2
(1 User control requirements differ according to update mode used . . .								S	II. II. 5. 1. 2. 2. 1
b. Data selection	H	H	H	H	H		S		II. II. 5. 2
(1) Achieved through statement of logical relations to be satisfied before the transaction is applied to the file								S	II. II. 5. 2. 1
(a) Conditional expressions used								S	II. II. 5. 2. 1. 1
(1 Capabilities described for conditional expressions listed under <u>Language Attributes of Self-Contained Systems</u> available to data selection process								S	II. II. 5. 2. 1. 1. 1
(2) Achieved through matching of transaction and entry identifiers with no further specification of data selection criteria								S	II. II. 5. 2. 2
c. Data change operations	H	H	H		H		S		II. II. 5. 3
(1) Data manipulation facilities described under Language Attributes of Self-Contained Systems available to data change process . . .								S	II. II. 5. 3. 1
(2) Arithmetic changes:	H	H	H				S		II. II. 5. 3. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(a) Use of literals; e.g., add (100, 000) to population where 100, 000 is a literal and population is a name								S	II. II. 5. 3. 2. 1
(b) Compuation of a new data value (the capability to compute a new data value from other data values)								S	II. II. 5. 3. 2. 2
(c) Arithmetic operators used include: *, =, +, /, **								S	II. II. 5. 3. 2. 3
(d) Arithmetic changes can be performed on value contained in all occurrences of a specific item								S	II. II. 5. 3. 2. 4
(e) Specific errors detected by the system during arithmetic change attempt								S	II. II. 5. 3. 2. 5
(f) Supplemental information reported by the system during arithmetic change attempt								S	II. II. 5. 3. 2. 6
(3) Non-arithmetic changes	H	H	H				S		II. II. 5. 3. 3
(a) Insert (the capability to insert physical values, sets of values, or entries to a file that has previously defined their logical counterparts of):								S	II. II. 5. 3. 3. 1
(1 Item:									II. II. 5. 3. 3. 1. 1
-a- Update mode(s) used								S	II. II. 5. 3. 3. 1. 1. 1
-b- Data change operators used .								S	II. II. 5. 3. 3. 1. 1. 2
-c- Specific errors reported by system during item level insert								S	II. II. 5. 3. 3. 1. 1. 3
-d- Supplemental information reported by system during item level insert								S	II. II. 5. 3. 3. 1. 1. 4
-e- User specified errors or information to be reported . . .								S	II. II. 5. 3. 3. 1. 1. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
-f- User specified course of action under predefined conditions								S	II. II. 5. 3. 3. 1. 1. 6
(2 Group:									II. II. 5. 3. 3. 1. 2
-a- Update mode(s) used								S	II. II. 5. 3. 3. 1. 2. 1
-b- Data change operators used .								S	II. II. 5. 3. 3. 1. 2. 2
-c- Specific errors reported by system during group level insert								S	II. II. 5. 3. 3. 1. 2. 3
-d- Supplemental information reported by system during group level insert								S	II. II. 5. 3. 3. 1. 2. 4
-e- User specified errors or information to be reported . . .								S	II. II. 5. 3. 3. 1. 2. 5
-f- User specified courses of action								S	II. II. 5. 3. 3. 1. 2. 6
(3 Entry									II. II. 5. 3. 3. 1. 3
-a- Update mode(s) used								S	II. II. 5. 3. 3. 1. 3. 1
-b- Data change operators used .								S	II. II. 5. 3. 3. 1. 3. 2
-c- Specific errors reported by system during entry level insert								S	II. II. 5. 3. 3. 1. 3 3
-d- Supplemental information reported by system during entry level insert								S	II. II. 5. 3. 3. 1. 3. 4
-e- User specified errors or information to be reported . . .								S	II. II. 5. 3. 3. 1. 3. 5
-f- User specified courses of action								S	II. II. 5. 3. 3. 1. 3. 6

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(b) Delete (the capability to delete physical values, sets of values or entries to a file that has previously defined their logical counterparts of:	H	H	H				S		II. II. 5. 3. 3. 2
(1 (Same as those listed under II-B-5-c-3-a)								S	II. II. 5. 3. 3. 2. 1
(c) Replace (the capability to replace physical values, sets of values or entries to a file that has previously defined their logical counterparts of:	H	H	H				S		II. II. 5. 3. 3. 3
(1 Same as those listed under II-B-5-c-3-a)								S	II. II. 5. 3. 3. 3. 1
(d) Identifier changes for:	H	H	H				S		II. II. 5. 3. 3. 4
(1 (Same as those listed under II-B-5-c-3 a)								S	II. II. 5. 3. 3. 4. 1
6. File update monitoring and error detection facilities		H	H		H		S		II. II. 6
a. System detected errors						S		S	II. II. 6. 1
b. Operating system detected errors						S		S	II. II. 6. 2
c. User specified conditions for error detection								S	II. II. 6. 3
d. On-line corrections permitted								S	II. II. 6. 4
e. Batched corrections permitted								S	II. II. 6. 5
f. Run history of recognized errors provided								S	II. II. 6. 6
g. List of rejected transactions provided								S	II. II. 6. 7
h. System statistics reported						S		S	II. II. 6. 8
i. Audit trail features provided						S		S	II. II. 6. 9
j. Backup file features provided								S	II. II. 6. 10
k. Maintenance of file storage during update:									II. II. 6. 11

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(1) Performed by operating system								S	II. II. 6. 11. 1
(2) Performed by the DMS								S	II. II. 6. 11. 2
C. Interrogation	H	H	H	H	H		S		II. III
1. Query language	H						S		II. III. 1
a. Sufficient repertoire of comparators and connectors is provided including:								S	II. III. 1. 1
(1) Operators						S		S	II. III. 1. 1. 1
(2) Logical connectors						S		S	II. III. 1. 1. 2
b. Expressions entail:									II. III. 1. 2
(1) Complex expressions						S		S	II. III. 1. 2. 1
(2) Levels of nesting						S		S	II. III. 1. 2. 2
(3) Combined expressions								S	II. III. 1. 2. 3
(4) Mixed arithmetic/boolean expressions								S	II. III. 1. 2. 4
c. Sufficient mathematical functions are provided								S	II. III. 1. 3
d. Procedural language actions:									II. III. 1. 4
(1) Open and close of files								S	II. III. 1. 4. 1
(2) Data transfer between levels of memory								S	II. III. 1. 4. 2
(3) Specification of arithmetic functions								S	II. III. 1. 4. 3
(4) Definition of temporary storage in high speed memory								S	II. III. 1. 4. 4
(5) Statement of labeling								S	II. III. 1. 4. 5
(6) Statements transferring execution control								S	II. III. 1. 4. 6
(7) Statement looping								S	II. III. 1. 4. 7
2. Data selection	H	H	H	H	H		S		II. III. 2
a. System facility used								S	II. III. 2. 1

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
b. Data selection statements available						S		S	II. III. 2. 2
c. Depth of search:									II. III. 2. 3
(1) Locate specified data element from any level in logical organization of the data base:								S	II. III. 2. 3. 1
(a) Retrieve from a specified instance of a repeating group								S	II. III. 2. 3. 1. 1
(b) Retrieve from all repetitions of a repeating group								S	II. III. 2. 3. 1. 2
(2) Distinguish between the absence or presence of data values at any level of data								S	II. III. 2. 3. 2
d. Capability to terminate a search based on a specified number of group instances qualified								S	II. III. 2. 4
e. Capability to use the results of a single selection as input values to a subsequent selection								S	II. III. 2. 5
f. Capability to locate groups of data that satisfy search criteria and allow the use of the following rules to define related data groups that qualify for retrieval:								S	II. III. 2. 6
(1) Locate only those instances of a group whose item values satisfy the search criteria								S	II. III. 2. 6. 1
(2) Locate a superior group only if all instances of a subordinate group satisfy the search criteria								S	II. III. 2. 6. 2
(3) Locate a superior group if at least one instance of a subordinate group satisfies the search criteria								S	II. III. 2. 6. 3
(4) Locate all subordinate groups of a qualified group								S	II. III. 2. 6. 4
(5) Locate a superior group based on a subordinate group that is more than one level removed								S	II. III. 2. 6. 5

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(6) Locate any group related to a group that satisfies the search criteria								S	II. III. 2. 6. 6
(7) Locate those instances of a group whose items do not satisfy the primary search criteria but satisfy alternate or secondary search criteria								S	II. III. 2. 6. 7
3. Data extract	H	H	H	H	H		S	S	II. III. 3
a. System facility used								S	II. III. 3. 1
b. Data extract statements available						S		S	II. III. 3. 2
c. Data elements extracted (data value)								S	II. III. 3. 3
d. Attributes of data element other than data value which can be extracted:									II. III. 3. 4
(1) Name								S	II. III. 3. 4. 1
(2) Type								S	II. III. 3. 4. 2
(3) Length								S	II. III. 3. 4. 3
(4) Date stamp								S	II. III. 3. 4. 4
(5) Existence status								S	II. III. 3. 4. 5
(6) Other descriptions stored at data definition level								S	II. III. 3. 4. 6
e. Extract data elements used as search criteria in conditional expression								S	II. III. 3. 5
f. Extract text through:									II. III. 3. 6
(1) Association with formatted data								S	II. III. 3. 6. 1
(2) Content based on search criteria								S	II. III. 3. 6. 2
g. Multiple routing of user-selected outputs								S	II. III. 3. 7
h. Optional routine or output to devices other than users terminal								S	II. III. 3. 8
i. Capability to extract particular sets of data values for additional processing (e.g., sorting) based on the evaluation of the conditional expression								S	II. III. 3. 9

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
4. Storage and Recalling of Interrogation Statements . . .	H	H	H	H	H		S		II. III. 4
a. Statements may be stored, recalled and modified for one use only								S	II. III. 4. 1
b. Statements may be stored, recalled, modified and restored under a new name for multiple uses, without destroying the original statement								S	II. III. 4. 2
c. Statements may be stored and recalled with the user supplying parameters to complete the query at execution time								S	II. III. 4. 3
d. Statements created for later execution can be composed:									II. III. 4. 4
(1) At a console								S	II. III. 4. 4. 1
(2) In a batch mode								S	II. III. 4. 4. 2
e. User can request execution of a pre-stored query directly from:									II. III. 4. 5
(1) Remote consoles								S	II. III. 4. 5. 1
(2) Local consoles								S	II. III. 4. 5. 2
f. Statements composed on-line are:.									II. III. 4. 6
(1) Executed directly								S	II. III. 4. 6. 1
(2) Batched for execution								S	II. III. 4. 6. 2
(3) Both								S	II. III. 4. 6. 3
5. Temporary file creation	H	H	H		H		S		II. III. 5
a. Establish a temporary file for additional processing which is a subset of and has the same logical structure as the file from which it was produced for additional processing								S	II. III. 5. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(1) Temporary file can be used as an input file in another function or output presentation . . .								S	II. III. 5. 1. 1
b. Establish a temporary file which is not the same logical or physical structure as the system file . .								S	II. III. 5. 2
(1) Temporary file can be used to produce additional copies of previous outputs or reports . .								S	II. III. 5. 2. 1
6. Report Formatting	H	H	H		H		S		II. III. 6
a. Standard mode - output presentation in selectable standard formats:	H	H	H		H		S		II. III. 6. 1
(1) Available under demand scheduling								S	II. III. 6. 1. 1
(2) Available under recurring scheduling (outputs produced at a specified frequency, e.g., daily, 3 times daily, weekly, etc.)								S	II. III. 6. 1. 2
b. Report generation mode - ability for user to compose desired output formats:	H	H	H		H		S		II. III. 6. 2
(1) Available under recurring scheduling								S	II. III. 6. 2. 1
(2) Available under demand scheduling								S	II. III. 6. 2. 2
c. User can specify:									II. III. 6. 3
(1) Recurring scheduling for standard mode output								S	II. III. 6. 3. 1
(2) Recurring scheduling for report generation mode output								S	II. III. 6. 3. 2
(3) Demand scheduling for standard mode output .								S	II. III. 6. 3. 3
(4) Demand scheduling for report generation mode output								S	II. III. 6. 3. 4
d. All validation and editing capabilities are available in									II. III. 6. 4

(TECHNIQUE: columns A–H)

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(1) Standard output mode								S	II. III. 6. 4. 1
(2) Report generation mode								S	II. III. 6. 4. 2
e. Output presentation capability includes as a library function the ability to compose:									II. III. 6. 5
(1) Cover pages								S	II. III. 6. 5. 1
(2) Title pages								S	II. III. 6. 5. 2
(3) Distribution lists								S	II. III. 6. 5. 3
(4) Special handling instructions								S	II. III. 6. 5. 4
f. On-line/off-line capabilities provided:		H	H	H					II. III. 6. 6
(1) On-line output media:		H	H	H					II. III. 6. 6. 1
(a) Typewriter								S	II. III. 6. 6. 1. 1
(b) Display								S	II. III. 6. 6. 1. 2
(c) Teletype								S	II. III. 6. 6. 1. 3
(2) Off-line output media:		H		H					II. III. 6. 6. 2
(a) Tape								S	II. III. 6. 6. 2. 1
(b) Disc								S	II. III. 6. 6. 2. 2
(c) Printer								S	II. III. 6. 6. 2. 3
(d) Card punch								S	II. III. 6. 6. 2. 4
(3) Audio:		H	H	H					II. III. 6. 6. 3
(a) Spelled voice								S	II. III. 6. 6. 3. 1
(b) Spoken voice								S	II. III. 6. 6. 3. 2
(4) Automatic conversion to account for the specific characteristics of any given device is provided								S	II. III. 6. 6. 4

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(a) Standard method for representing characters that do not exist on the device is included								S	II. III. 6. 6. 4. 1
7. Standard output mode features		H	H		H		S		II. III. 7
a. Standard formats are:									II. III. 7. 1
(1) Parameter-driven								S	II. III. 7. 1. 1
(2) Invoked by name								S	II. III. 7. 1. 2
b. Standard report set includes the following report types:									II. III. 7. 2
(1) Tabular-column presentation of data								S	II. III. 7. 2. 1
(2) Row-oriented (a simple row-by-row list of data values with indentation to indicate their position in the hierarchy								S	II. III. 7. 2. 2
(3) Functional formats:									II. III. 7. 2. 3
(a) Presentation of counts								S	II. III. 7. 2. 3. 1
(b) Summations								S	II. III. 7. 2. 3. 2
(4) File format (a presentation of the names and relationships of the levels of the file).								S	II. III. 7. 2. 4
c. Automatic formatting capabilities include:									II. III. 7. 3
(1) Column/row width								S	II. III. 7. 3. 1
(2) Column/row spacing								S	II. III. 7. 3. 2
(3) Line width								S	II. III. 7. 3. 3
(4) Number of lines per page						S		S	II. III. 7. 3. 4
(5) Above parameters can be automatically calculated and adjusted to the output device capacity								S	II. III. 7. 3. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(6) User can specify parameters for the above formats as an override option								S	II. III. 7. 3. 6
d. User can specify the following parameters when initiating standard formats:									II. III. 7. 4
(1) Report title(s)								S	II. III. 7. 4. 1
(2) Security classification								S	II. III. 7. 4. 2
(a) User authorization								S	II. III. 7. 4. 2. 1
(b) Highest classification of output data								S	II. III. 7. 4. 2. 2
(3) Date or as-of-time								S	II. III. 7. 4. 3
(4) Page numbers								S	II. III. 7. 4. 4
(5) Column headings:									II. III. 7. 4. 5
(a) Item name (item name appears at head of column of its values)								S	II. III. 7. 4. 5. 1
(b) Query specified (user specified title replacing or added to the item name)								S	II. III. 7. 4. 5. 2
(c) Data description title (heading other than item name specified in the data description)								S	II. III. 7. 4. 5. 3
(6) Trailer information (information provided at bottom of each output page; e.g., page number and security classification)								S	II. III. 7. 4. 6
(7) Table of contents based on report titles								S	II. III. 7. 4. 7
e. Special functions which can be selected by user:									II. III. 7. 5
(1) Counts of item occurrences								S	II. III. 7. 5. 1
(2) Counts of unique item values								S	II. III. 7. 5. 2
(3) Counts of particular item values								S	II. III. 7. 5. 3
(4) Counts based on conditional queries								S	II. III. 7. 5. 4

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(5) Sums of particular item values								S	II. III. 7. 5. 5
(6) Percent total								S	II. III. 7. 5. 6
(7) Different reports (system can generate different report formats from one retrieval statement)								S	II. III. 7. 5. 7
(8) Multiple copies (system can provide more than one original copy of the same report)								S	II. III. 7. 5. 8
(9) Statistical functions of item values									II. III. 7. 5. 9
(a) Mean								S	II. III. 7. 5. 9. 1
(b) Median								S	II. III. 7. 5. 9. 2
(c) Mode								S	II. III. 7. 5. 9. 3
(d) Standard deviation								S	II. III. 7. 5. 9. 4
(10) Editing functions								S	II. III. 7. 5. 10
(11) Decoding functions								S	II. III. 7. 5. 11
(12) Capability to override decoding transformations (obtain stored data value)								S	II. III. 7. 5. 12
(13) Subtotals								S	II. III. 7. 5. 13
(14) Subcounts								S	II. III. 7. 5. 14
3. Report Generation Mode Features		H	H		H		S		II. III. 8
a. Headers - the capability to employ for titles, row headers and column headers, the following:								S	II. III. 8. 1
(1) Literal values								S	II. III. 8. 1. 1
(2) Item names								S	II. III. 8. 1. 2
b. Positioning - the capability to specify for titles, column headings, row titles and items, the following:								S	II. III. 8. 2
(1) Vertical position								S	II. III. 8. 2. 1
(2) Horizontal position								S	II. III. 8. 2. 2
(3) Data value position								S	II. III. 8. 2. 3

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(4) Right justification								S	II. III. 8. 2. 4
(5) Left justification								S	II. III. 8. 2. 5
c. Pagination specification - the capability for the user to specify the following parameters:									II. III. 8. 3
(1) Starting page								S	II. III. 8. 3. 1
(2) Page increment number								S	II. III. 8. 3. 2
(3) Page number								S	II. III. 8. 3. 3
(4) Last page								S	II. III. 8. 3. 4
(5) Location of page number								S	II. III. 8. 3. 5
(6) Page break on:									II. III. 8. 3. 6
(a) Sort key								S	II. III. 8. 3. 6. 1
(b) Value of a sum								S	II. III. 8. 3. 6. 2
(c) Number of lines on a page								S	II. III. 8. 3. 6. 3
(d) Particular count value								S	II. III. 8. 3. 6. 4
(e) Output form								S	II. III. 8. 3. 6. 5
(7) Termination of output based on:									II. III. 8. 3. 7
(a) Number of pages								S	II. III. 8. 3. 7. 1
(b) Number of lines								S	II. III. 8. 3. 7. 2
(c) Item values								S	II. III. 8. 3. 7. 3
(8) In the absence of these specifications, several standard options are provided						S		S	II. III. 8. 3. 8
d. Data reduction features include:									II. III. 8. 4
(1) Maximum item value								S	II. III. 8. 4. 1
(2) Minimum item value								S	II. III. 8. 4. 2
(3) Statistical functions of specified item values:									II. III. 8. 4. 3
(a) Mean								S	II. III. 8. 4. 3. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(b) Median								S	II. III. 8. 4. 3. 2
(c) Mode								S	II. III. 8. 4. 3. 3
(d) Standard deviation								S	II. III. 8. 4. 3. 4
(4) Counts of item occurrences								S	II. III. 8. 4. 4
(5) Counts of all unique item values								S	II. III. 8. 4. 5
(6) Counts of particular item values								S	II. III. 8. 4. 6
(7) Sums of all occurrences of an item								S	II. III. 8. 4. 7
(8) Percent total								S	II. III. 8. 4. 8
e. Page headers and trailers include:									II. III. 8. 5
(1) Security classification								S	II. III. 8. 5. 1
(2) Page number								S	II. III. 8. 5. 2
(3) Date								S	II. III. 8. 5. 3
(4) Time								S	II. III. 8. 5. 4
(5) Report titles								S	II. III. 8. 5. 5
(6) Table of contents								S	II. III. 8. 5. 6
(7) Column headings								S	II. III. 8. 5. 7
(a) Item name/number								S	II. III. 8. 5. 7. 1
(b) Title specified in query								S	II. III. 8. 5. 7. 2
(c) Title for column heading in data description								S	II. III. 8. 5. 7. 3
f. Outputs									II. III. 8. 6
(1) Use of preprinted forms								S	II. III. 8. 6. 1
(2) Spread sheet output (e. g., output is 2 physical pages wide)								S	II. III. 8. 6. 2
(3) Different report formats from one retrieval								S	II. III. 8. 6. 3
(4) Multiple copies of a report from one retrieval								S	II. III. 8. 6. 4
g. Special outputs available upon request:								S	II. III. 8. 7

The columns A through H are headed TECHNIQUE.

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFFRENCE NO
(1) Control file reports:								S	II. III. 8. 7. 1
(a) List of stored procedures								S	II. III. 8. 7. 1. 1
(b) Specific logical file organization								S	II. III. 8. 7. 1. 2
(c) Items in a specified group								S	II. III. 8. 7. 1. 3
(d) Data definitions of:								S	II. III. 8. 7. 1. 4
(1) Coding								S	II. III. 8. 7. 1. 4. 1
(2) Transformation								S	II. III. 8. 7. 1. 4. 2
(3) Validation criteria								S	II. III. 8. 7. 1. 4. 3
(4) Security								S	II. III. 8. 7. 1. 4. 4
(2) Job summaries:									II. III. 8. 7. 2
(a) Rejected data								S	II. III. 8. 7. 2. 1
(b) Validated transaction data								S	II. III. 8. 7. 2. 2
(c) Invoked procedures								S	II. III. 8. 7. 2. 3
(d) Error conditions								S	II. III. 8. 7. 2. 4
(e) Statistics (e.g., number of entries processed, volume of transaction data, etc.)								S	II. III. 8. 7. 2. 5
D. Language Attributes of Self-Contained Systems									II. IV
1. Language form		H	H				S		II. IV. 1
a. Narrative								S	II. IV. 1. 1
b. Keyword								S	II. IV. 1. 2
c. Separator								S	II. IV. 1. 3
d. Fixed position								S	II. IV. 1. 4
e. User defined changes to the language:									II. IV. 1. 5
(1) Synonyms for verbs								S	II. IV. 1. 5. 1
(2) Syntax changes								S	II. IV. 1. 5. 2
(3) Semantic changes								S	II. IV. 1. 5. 3

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
2. Language Operands		H	H				S		II. IV. 2
a. Simple operands:								S	II. IV. 2. 1
(1) Item								S	II. IV. 2. 1. 1
(2) String value								S	II. IV. 2. 1. 2
(3) Numeric value								S	II. IV. 2. 1. 3
(4) Variable								S	II. IV. 2. 1. 4
(5) Functions								S	II. IV. 2. 1. 5
(6) Square root								S	II. IV. 2. 1. 6
(7) Natural logarithm								S	II. IV. 2. 1. 7
(8) Results of computation								S	II. IV. 2. 1. 8
(9) Trigonometric:								S	II. IV. 2. 1. 9
(a) Sine								S	II. IV. 2. 1. 9. 1
(b) Cosine								S	II. IV. 2. 1. 9. 2
(c) Tangent								S	II. IV. 2. 1. 9. 3
(d) Arcsine, arccosine, arctangent								S	II. IV. 2. 1. 9. 4
b. Compound operands:								S	II. IV. 2. 2
(1) Any combination of simple operands								S	II. IV. 2. 2. 1
(2) Different group instance; same item								S	II. IV. 2. 2. 2
(3) Different entry; same item								S	II. IV. 2. 2. 3
c. Transformation of standard operands:									II. IV. 2. 3
(1) System can transform the following operands to present them consistently with their file counterparts:									II. IV. 2. 3. 1
(a) Date								S	II. IV. 2. 3. 1. 1
(b) Time								S	II. IV. 2. 3. 1. 2
(c) Temperature								S	II. IV. 2. 3. 1. 3
(d) Location								S	II. IV. 2. 3. 1. 4
(e) Pressure								S	II. IV. 2. 3. 1. 5

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERFNCE NO
(f) Distance								S	II. IV. 2. 3. 1. 6
(g) Volume								S	II. IV. 2. 3. 1. 7
3. Language Operators		H	H				S		II. IV. 3
a. Basic relational operators:									II. IV. 3. 1
(1) Equal								S	II. IV. 3. 1. 1
(2) Not equal								S	II. IV. 3. 1. 2
(3) Greater than								S	II. IV. 3. 1. 3
(4) Less than								S	II. IV. 3. 1. 4
(5) Greater than or equal								S	II. IV. 3. 1 5
(6) Less than or equal								S	II. IV. 3. 1. 6
b. Special operators:									II. IV. 3. 2
(1) Greater than but not blank								S	II. IV. 3. 2. 1
(2) Less than but not blank								S	II. IV. 3. 2. 2
(3) Character pattern								S	II. IV. 3. 2. 3
(4) Absence								S	II. IV. 3. 2. 4
(5) Between								S	II. IV. 3. 2. 5
(6) Mask match								S	II. IV. 3. 2. 6
(7) Maximum/minimum								S	II. IV. 3. 2. 7
(8) Empty								S	II. IV. 3. 2. 8
(9) Increase/decrease								S	II. IV. 3. 2. 9
c. Functional operators:									II. IV. 3. 3
(1) Increase								S	II. IV. 3. 3. 1
(2) Decrease								S	II. IV. 3. 3. 2
(3) Product								S	II. IV. 3. 3. 3
(4) Sum								S	II. IV. 3. 3. 4
d. Arithmetic operators:									II. IV. 3. 4

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(1) =								S	II. IV. 3. 4. 1
(2) +								S	II. IV. 3. 4. 2
(3) -								S	II. IV. 3. 4. 3
(4) /								S	II. IV. 3. 4. 4
(5) *								S	II. IV. 3. 4. 5
(6) **								S	II. IV. 3. 4. 6
e. Mode of computation permitted:									II. IV. 3. 5
(1) Floating decimal								S	II. IV. 3. 5. 1
(2) Decimal								S	II. IV. 3. 5. 2
(3) Integer								S	II. IV. 3. 5. 3
f. Reduction operators:									II. IV. 3. 6
(1) Count								S	II. IV. 3. 6. 1
(2) Total								S	II. IV. 3. 6. 2
(3) Average								S	II. IV. 3. 6. 3
(4) Partials (subtotals, subcounts)								S	II. IV. 3. 6. 4
g. Logical connectors:									II. IV. 3. 7
(1) AND								S	II. IV. 3. 7. 1
(2) Exclusive OR								S	II. IV. 3. 7. 2
(3) Inclusive OR								S	II. IV. 3. 7. 3
(4) NOT								S	II. IV. 3. 7. 4
h. Geographic search:									II. IV. 3. 8
(1) Circle								S	II. IV. 3. 8. 1
(2) Convex polygon								S	II. IV. 3. 8. 2
(3) Concave polygon								S	II. IV. 3. 8. 3
(4) Straight-line intersections								S	II. IV. 3. 8. 4
(5) Combinations of the above								S	II. IV. 3. 8. 5
4. Statistical Operations		H	H		H		S		II. IV. 4

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
a. Calculate:									II. IV. 4. 1
(1) Arithmetic mean								S	II. IV. 4. 1. 1
(2) Mode								S	II. IV. 4. 1. 2
(3) Median								S	II. IV. 4. 1. 3
(4) Standard deviation of a field								S	II. IV. 4. 1. 4
b. Count:									II. IV. 4. 2
(1) Number of unique values of a field								S	II. IV. 4. 2. 1
(2) Number of occurrences of a particular field value								S	II. IV. 4. 2. 2
5. Conditional Expressions		H	H		H		S		II. IV. 5
a. Types of conditional expressions permitted									II. IV. 5. 1
(1) Logical								S	II. IV. 5. 1. 1
(2) Arithmetic								S	II. IV. 5. 1. 2
(3) Boolean								S	II. IV. 5. 1. 3
(4) Combination of the above								S	II. IV. 5. 1. 4
b. Natural evaluation of expressions (e. g., left to right scan of the following symbols: (), *, -, +, and exponentiation)								S	II. IV. 5. 2
c. Complexity of conditional expressions:									II. IV. 5. 3
(1) Capability provided to mix arithmetic and boolean expressions								S	II. IV. 5. 3. 1
(2) Operand may be mixed mode								S	II. IV. 5. 3. 2
(3) Number of conditional expressions that can be combined directly						S		S	II. IV. 5. 3. 3
(4) Number of levels of nesting using parentheses						S		S	II. IV. 5. 3. 4
(5) Minimum of 40 operators and operands per expression								S	II. IV. 5. 3. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(6) Compound conditions on same item within expression								S	II. IV. 5. 3. 6
(7) Precedence rules for logical connectors within expression						S		S	II. IV. 5. 3. 7
(8) Logically connected conditions on several items but with the same reference quantity within expression								S	II. IV. 5. 3. 8
d. Conditions may be specified for:									II. IV. 5. 4
(1) Items								S	II. IV. 5. 4. 1
(2) Groups								S	II. IV. 5. 4. 2
(3) Entries								S	II. IV. 5. 4. 3
e. Expression can be given an identifier								S	II. IV. 5. 5
III. DMS System Control		H	H		H		S	H	III
A. Monitoring				H	H				III. I
1. Recording Control									III. I. 1
a. Capability for user control (user can specify types of events to be recorded).						S		S	III. I. 1. 1
b. Number of recording categories provided								S	III. I. 1. 2
c. Capability for system to vary the recording frequency based on:									III. I. 1. 3
(1) Time of day				H	H			S	III. I. 1. 3. 1
(2) Time interval				H	H			S	III. I. 1. 3. 2
(3) Sampling rate				H	H			S	III. I. 1. 3. 3
(4) Sampling period				H	H			S	III. I. 1. 3. 4
(5) System recording category in operation . . .				H	H			S	III. I. 1. 3. 5
2. General Information Recorded									III. I. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
a. Capability for generating and recording the following types of statistics:									III. I. 2. 1
(1) System tallies:									III. I. 2. 1. 1
(a) Item retrieved - count of the number of items retrieved					H			S	III. I. 2. 1. 1. 1
(b) System module executed - a list of which system modules have been executed and how often				H	H			S	III. I. 2. 1. 1. 2
(c) Device usage - tallies of the number of consoles in use and the number of disc seeks issued				H	H			S	III. I. 2. 1. 1. 3
(d) Input transactions - a tally of the number of transactions applied to a file and of those transactions which were rejected					H			S	III. I. 2. 1. 1. 4
(2) System event times:									III. I. 2. 1. 2
(a) Job execution - total time for processing a specific job	H			H	H			S	III. I. 2. 1. 2. 1
(b) System module execution - total processing time for each system module used	H			H	H			S	III. I. 2. 1. 2. 2
(c) Data access - total time required to search the file	H			H	H			S	III. I. 2. 1. 2. 3
(3) Job history:									III. I. 2. 1. 3
(a) Capability to provide a history of all DMS jobs on demand					H			S	III. I. 2. 1. 3. 1
(b) History information includes:									III. I. 2. 1. 3. 2
(1 Programmer definition								S	III. I. 2. 1. 3. 2. 1
(2 Job time:								S	III. I. 2. 1. 3. 2. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
-a- Time on								S	III. I. 2. 1. 3. 2. 2. 1
-b- Time off								S	III. I. 2. 1. 3. 2. 2. 2
-c- Duration								S	III. I. 2. 1. 3. 2. 2. 3
(3 Job termination									III. I. 2. 1. 3. 2. 3
-a- Successful								S	III. I. 2. 1. 3. 2. 3. 1
-b- Unsuccessful								S	III. I. 2. 1. 3. 2. 3. 2
-c- Abnormal termination reason								S	III. I. 2. 1. 3. 2. 3. 3
(4 I/O time by I/O event				H	H			S	III. I. 2. 1. 3. 2. 4
(5 CPU time by task				H				S	III. I. 2. 1. 3. 2. 5
(6 DMS modules executed by execution sequence					H			S	III. I. 2. 1. 3. 2. 6
(7 O/S modules executed by execution sequence				H	H			S	III. I. 2. 1. 3. 2. 7
(8 I/O device usage by access sequence				H	H			S	III. I. 2. 1. 3. 2. 8
(9 Data accessed or modified					H			S	III. I. 2. 1. 3. 2. 9
(10 Record of operator intervention					H			S	III. I. 2. 1. 3. 2. 10
(c) Capability for off-line storage of historical data								S	III. I. 2. 1. 3. 3
3. Specific Information Recorded								S	III. I. 3
a. Data base access								S	III. I. 3. 1
(1) Capability to record for any level of data, the following:									III. I. 3. 1. 1
(a) Who initiated the access								S	III. I. 3. 1. 1. 1
(b) What program modules were employed, including, for each employment:								S	III. I. 3. 1. 1. 2
(1 Time in					H			S	III. I. 3. 1. 1. 2. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(2 Time out					H			S	III. I. 3. 1. 1. 2. 2
(c) What level of data was accessed					H			S	III. I. 3. 1. 1. 3
(2) Capability to define new data to be collected								S	III. I. 3. 1. 2
(3) File backup capability for recovery purposes or statistical analysis								S	III. I. 3. 1. 3
b. Program modules								S	III. I. 3. 2
(1) Capability to obtain for each program module used:									III. I. 3. 2. 1
(a) User identification					H			S	III. I. 3. 2. 1. 1
(b) Initiation time				H				S	III. I. 3. 2. 1. 2
(c) Termination time				H				S	III. I. 3. 2. 1. 3
(d) Other information as needed								S	III. I. 3. 2. 1. 4
4. Monitoring Levels									III. I. 4
a. Provision for demand monitoring								S	III. I. 4. 1
b. Provision for background monitoring on a continuing basis								S	III. I. 4. 2
(1) Demand monitoring:									III. I. 4. 2. 1
(a) Standard queries:									III. I. 4. 2. 1. 1
(1 What queries are currently active					H			S	III. I. 4. 2. 1. 1. 1
(2 What type of file is being accessed								S	III. I. 4. 2. 1. 1. 2
(3 What is the classification of the file being accessed								S	III. I. 4. 2. 1. 1. 3
(4 Amount of working storage available				H	H			S	III. I. 4. 2. 1. 1. 4
(5 Which devices are currently in use				H	H			S	III. I. 4. 2. 1. 1. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
(b) Specialized queries:									III. I. 4. 2. 1. 2
(1 Capability to specify:									III. I. 4. 2. 1. 2. 1
-a- Which events are to be recorded								S	III. I. 4. 2. 1. 2. 1. 1
-b- Conditional recording of events								S	III. I. 4. 2. 1. 2. 1. 2
-c- Dynamic overriding of recording								S	III. I. 4. 2. 1. 2. 1. 3
(2 Specialized queries can include:									III. I. 4. 2. 1. 2. 2
-a- Itemization of all users who have accessed a specific file within a given time period								S	III. I. 4. 2. 1. 2. 2. 1
-b- Time of last update performed on a specific file								S	III. I. 4. 2. 1. 2. 2. 2
-c- Person who performed the update								S	III. I. 4. 2. 1. 2. 2. 3
-d- Data accessed by a specific user within a given time period								S	III. I. 4. 2. 1. 2. 2. 4
-e- A list of all files currently in the system and their classification, size and resident storage device								S	III. I. 4. 2. 1. 2. 2. 5
-f- A list of all users authorized access at a given security level and all files included in this classification								S	III. I. 4. 2. 1. 2. 2. 6
-g- Itemization by name of all procedures created by a specified user, providing them for selective viewing on a CRT terminal								S	III. I. 4. 2. 1. 2. 2. 7
(2) Background monitoring									III. I. 4. 2. 2

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(a) Capability for system manager to monitor the system for instances of abnormal system use:								S	III. I. 4. 2. 2. 1
(1 Attempts to access system operation capabilities				H				S	III. I. 4. 2. 2. 1. 1
(2 Attempts to access the data base illegally				H				S	III. I. 4. 2. 2. 1. 2
(3 Abnormal termination of runs . . .				H				S	III. I. 4. 2. 2. 1. 3
B. Error Recording		H	H	H	H	S	S		III. II
1. System Detected Errors					H				III. II. 1
a. Item values - includes invalid characters, incorrect number of characters, invalid values, etc. .								S	III. II. 1. 1
b. Transaction format - includes invalid field lengths, incorrect sequence of values, invalid transaction codes, etc.								S	III. II. 1. 2
c. Procedural statements - involves syntax or punctuation of procedural statements								S	III. II. 1. 3
2. Operating System Detected Errors				H					III. II. 2
a. Equipment malfunctions - hardware generated errors								S	III. II. 2. 1
b. Task or job specification errors - errors in requesting the execution of a task or job, e. g., incorrectly naming the program to be executed, incorrect device specification, etc.								S	III. II. 2. 2
3. Diagnostics Provided									III. II. 3
a. Snapshot dumps of work areas								S	III. II. 3. 1
b. System program dumps								S	III. II. 3. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
c. Breakpoint capability:									III. II. 3. 3
(1) Conditional termination							S	S	III. II. 3. 3. 1
(2) Conditional execution							S	S	III. II. 3. 3. 2
d. Trace of the procedural program								S	III. II. 3. 4
e. Indication when a file or procedure is not acceptable for incorporation into the system					H			S	III. II. 3. 5
f. Trace of item values at any level of aggregation which is being accessed					H			S	III. II. 3. 6
4. User Specifications for Dumps and Traces									III. II. 4
a. User specified output media for dump or trace								S	III. II. 4. 1
b. User can cause dumps to occur or traces to begin and end points within the processing which are:									III. II. 4. 2
(1) Predefined					H			S	III. II. 4. 2. 1
(2) Conditionally defined (when procedure is compiled)					H			S	III. II. 4. 2. 2
(3) Defined at execution time for batch mode					H			S	III. II. 4. 2. 3
(4) Dynamically defined on-line					H			S	III. II. 4. 2. 4
c. User specified amount of output from a dump or trace								S	III. II. 4. 3
d. User specified type of information that is made available for each element of the dump or trace								S	III. II. 4. 4
e. User specified memory areas to be dumped:									III. II. 4. 5
(1) Absolute memory locations				H	H			S	III. II. 4. 5. 1
(2) Symbolic names of:									III. II. 4. 5. 2
(a) Jobs					H			S	III. II. 4. 5. 2. 1
(b) Procedures					H			S	III. II. 4. 5. 2. 2
(3) Parts of jobs or procedures					H			S	III. II. 4. 5. 3

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(4) Several non-contiguous areas of memory by a single dump command				H	H			S	III. II. 4. 5. 4
f. User specified data base areas to be dumped:									III. II. 4. 6
(1) File names								S	III. II. 4. 6. 1
(2) Group names								S	III. II. 4. 6. 2
(3) Item names								S	III. II. 4. 6. 3
(4) Several portions of a data base in a single dump statement, e. g., two different files								S	III. II. 4. 6. 4
g. User specified traces of:									III. II. 4. 7
(1) Procedures and subroutines								S	III. II. 4. 7. 1
(2) Times:								S	III. II. 4. 7. 2
(a) Start				H			S	S	III. II. 4. 7. 2. 1
(b) End				H			S	S	III. II. 4. 7. 2. 2
(c) Duration				H			S	S	III. II. 4. 7. 2. 3
(3) Input arguments:									III. II. 4. 7. 3
(a) Names							S	S	III. II. 4. 7. 3. 1
(b) Values							S	S	III. II. 4. 7. 3. 2
(4) Items updated:									III. II. 4. 7. 4
(a) Item identification							S	S	III. II. 4. 7. 4. 1
(b) Old item value							S	S	III. II. 4. 7. 4. 2
(c) New item value							S	S	III. II. 4. 7. 4. 3
(d) Time of update							S	S	III. II. 4. 7. 4. 4
(5) System resources used:									III. II. 4. 7. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(a) Type of resource, e.g., CPU component, tape unit							S	S	III. II. 4. 7. 5. 1
(b) Specific identification of resource							S	S	III. II. 4. 7. 5. 2
h. User specified limitation of trace of:									III. II. 4. 8
(1) Procedures							S	S	III. II. 4. 8. 1
(2) Data							S	S	III. II. 4. 8. 2
(3) Resources							S	S	III. II. 4. 8. 3
i. User specified limitation of trace to:									III. II. 4. 9
(1) Specified occurrence intervals					H		S	S	III. II. 4. 9. 1
(2) Specified logical conditions					H		S	S	III. II. 4. 9. 2
(3) Specified maximum trace lengths					H		S	S	III. II. 4. 9. 3
(4) Specified points for trace to begin or end in terms of:									III. II. 4. 9. 4
(a) Occurrence intervals; e.g., begin trace after 50th subroutine has been executed				H	H		S	S	III. II. 4. 9. 4. 1
(b) Times; e.g., cease trace after 10 minutes				H	H		S	S	III. II. 4. 9. 4. 2
C. Restart and Recovery		H	H						III. III
1. Detection and recovery provided from errors caused by:									III. III. 1
a. Programmer							S	S	III. III. 1. 1
b. Operator				H			S	S	III. III. 1. 2
c. Hardware				H			S	S	III. III. 1. 3
d. System				H			S	S	III. III. 1. 4
2. Error messages contain recovery instructions automatically or upon request							S	S	III. III. 2
3. File backup capability for restart or recovery from data base damage							S	S	III. III. 3

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
a. Generation of log file of transactions entering system from terminals							S	S	III. III. 3. 1
b. Generation of log file of messages sent by system to terminal							S	S	III. III. 3. 2
c. Storage needs for backup, e.g., extra tape drives.							S	S	III. III. 3. 3
4. Processing interrupt:									III. III. 4
a. Processing may be abandoned at any time by the user							S	S	III. III. 4. 1
b. Processing of a user program may be suspended at any time by:									III. III. 4. 2
(1) The system operator							S	S	III. III. 4. 2. 1
(2) The user							S	S	III. III. 4. 2. 2
c. Recovery from job suspension provided through:									III. III. 4. 3
(1) Full save provisions								S	III. III. 4. 3. 1
(2) Full restart provisions								S	III. III. 4. 3. 2
d. Processing can be terminated by user at other than normal termination points							S	S	III. III. 4. 4
e. Portions of a program normally processed can be skipped on a temporary (one time only) basis							S	S	III. III. 4. 5
D. Security Features		H	H		H		S		III. IV
1. Security restrictions can be applied at file definition time at:									III. IV. 1
a. File level								S	III. IV. 1. 1
b. Entry level								S	III. IV. 1. 2
c. Group level								S	III. IV. 1. 3
d. Item level								S	III. IV. 1. 4

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
2. Security restriction applies to:									III. IV. 2
a. Data access								S	III. IV. 2. 1
b. Data modification								S	III. IV. 2. 2
3. Clearance takes place:									III. IV. 3
a. At the opening of the data base								S	III. IV. 3. 1
b. At time of issuing data handling language statements								S	III. IV. 3. 2
4. Clearance levels provided for any level of data:									III. IV. 4
a. Unclassified								S	III. IV. 4. 1
b. Confidential								S	III. IV. 4. 2
c. Secret								S	III. IV. 4. 3
d. Top secret								S	III. IV. 4. 4
5. Number of access categories within each security level						S		S	III. IV. 5
6. Specification of a higher classification for an entire report than the classification of the report components								S	III. IV. 6
7. Capability to have classification level names adjusted by the user								S	III. IV. 7
8. Capability to have classification level names abbreviated								S	III. IV. 8
9. Provision of an automatic destruction capability for: .									III. IV. 9
a. All on-line storage devices							S	S	III. IV. 9. 1
b. Remote disc packs							S	S	III. IV. 9. 2
c. Remote tapes							S	S	III. IV. 9. 3
10. Necessitation of physical intervention for destruction of:									III. IV. 10

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
a. Internal storage								S	III. IV. 10. 1
b. Non-internal storage, e.g., tape, disc								S	III. IV. 10. 2
11. Provision of read protection for:									III. IV. 11
a. File							S	S	III. IV. 11. 1
b. Entry							S	S	III. IV. 11. 2
c. Group							S	S	III. IV. 11. 3
d. Item							S	S	III. IV. 11. 4
12. Provision of write protection for a:									III. IV. 12
a. File							S	S	III. IV. 12. 1
b. Entry							S	S	III. IV. 12. 2
c. Group							S	S	III. IV. 12. 3
d. Item							S	S	III. IV. 12. 4
IV. Host Environment Interrelationships		H	H	H	H		S		IV
A. Host Language Attributes									IV. I
1. Levels of interface provided									IV. I. 1
a. Narrative or free form of writing down manipulation language statements								S	IV. I. 1. 1
b. Fixed tabular format (less scanning is required).								S	IV. I. 1. 2
c. Call processing module (go directly to processing module, no scanning, parsing, validation or interpretation of data manipulation language statement is required)								S	IV. I. 1. 3
d. Machine oriented interface (statement reference data at the detailed physical I/O level)								S	IV. I. 1. 4
2. Programming modes		H	H				S		IV. I. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
a. Input (the transferral of data from the data base to the program)								S	IV. I. 2. 1
b. Output (the transferral of data from program to data base. Used in file creation or add modification)								S	IV. I. 2. 2
c. Update (encompasses input and output):								S	IV. I. 2. 3
(1) System keeps track of generations of a file								S	IV. I. 2. 3. 1
d. Access mode restrictions (i.e., update can only be performed on random file under COBOL)								S	IV. I. 2. 4
3. Access methods	H	H	H	H	H		S		IV. I. 3
a. Sequential	H	H	H	H	H		S	S	IV. I. 3. 1
b. Random	H	H	H	H	H		S	S	IV. I. 3. 2
c. User specified	H	H	H	H	H		S	S	IV. I. 3. 3
d. No mode distinction made by the user								S	IV. I. 3. 4
4. Method of invoking programming facilities									IV. I. 4
a. Programming facilities can be invoked from any host language								S	IV. I. 4. 1
b. Method of interface is different from one host language to another								S	IV. I. 4. 2
c. Method of invocation:									IV. I. 4. 3
(1) Call to control module								S	IV. I. 4. 3. 1
(2) Verb set								S	IV. I. 4. 3. 2
d. Explicit exit required from host language								S	IV. I. 4. 4
5. Language form:							S		IV. I. 5
a. Narrative								S	IV. I. 5. 1

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
b. Fixed position								S	IV. I. 5. 2
c. Separator								S	IV. I. 5. 3
d. Keyword								S	IV. I. 5. 4
6. Addressable data structures									IV. I. 6
a. File:								S	IV. I. 6. 1
(1) Referencing statement							S	S	IV. I. 6. 1. 1
b. Entry:								S	IV. I. 6. 2
(1) Referencing statement							S	S	IV. I. 6. 2. 1
c. Group:								S	IV. I. 6. 3
(1) Referencing statement							S	S	IV. I. 6. 3. 1
d. Item:								S	IV. I. 6. 4
(1) Referencing statement							S	S	IV. I. 6. 4. 1
7. Program-system communication		H	H						IV. I. 7
a. Currency (keeping track of current position in the data base):									IV. I. 7. 1
(1) Single pointer								S	IV. I. 7. 1. 1
(2) Multiple pointers								S	IV. I. 7. 1. 2
(3) User must reset all currency pointers in an update								S	IV. I. 7. 1. 3
(4) Pointers are maintained in terms of an internal identifier for each group								S	IV. I. 7. 1. 4
(5) Pointers reference:									IV. I. 7. 1. 5

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(a) Entries								S	IV. I. 7. 1. 5. 1
(b) Current group (most-recent group processed)								S	IV. I. 7. 1. 5. 2
(c) Parent group of current group								S	IV. I. 7. 1. 5. 3
b. Error handling						S	S	S	IV. I. 7. 2
c. Selection criteria:									IV. I. 7. 3
(1) Selection of data is accomplished by the association of an identifier with the data manipulation language statement							S	S	IV. I. 7. 3. 1
(2) Selection of data is accomplished by the association of a conditional expression with the data manipulation language statement:							S	S	IV. I. 7. 3. 2
(a) Conditional expression capabilities for selection						S	S	S	IV. I. 7. 3. 2. 1
(3) Form and content of selection criteria							S		IV. I. 7. 3. 3
(a) Logical and relational operators						S	S	S	IV. I. 7. 3. 3. 1
(b) Comparison of item values to:									IV. I. 7. 3. 3. 2
(1 Constants							S	S	IV. I. 7. 3. 3. 2. 1
(2 Ranges							S	S	IV. I. 7. 3. 3. 2. 2
(3 Other item values							S	S	IV. I. 7. 3. 3. 2. 3
(c) Existence conditions							S	S	IV. I. 7. 3. 3. 3
(4) Selection criteria appear in the data manipulation language statement							S	S	IV. I. 7. 3. 4
(5) Selection is achieved through communication items initialized by the user with values of identifier items								S	IV. I. 7. 3. 5

The columns A–H fall under the heading TECHNIQUE.

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
8. Security									IV. I. 8
a. Security clearance takes place:									IV. I. 8. 1
(1) At time of opening part of data base for processing							S	S	IV. I. 8. 1. 1
(2) At time of issuing data manipulation statement							S	S	IV. I. 8. 1. 2
b. Program is linked to only part of the data base in a binding process; the privacy of the rest of the data base is automatically ensured							S	S	IV. I. 8. 2
c. Security restriction is applied to:									IV. I. 8. 3
(1) Data modification							S	S	IV. I. 8. 3. 1
(2) Data access							S	S	IV. I. 8. 3. 2
(3) Both							S	S	IV. I. 8. 3. 3
d. Security is based on:									IV. I. 8. 4
(1) Authority level								S	IV. I. 8. 4. 1
(2) Need-to-know level								S	IV. I. 8. 4. 2
(3) Both								S	IV. I. 8. 4. 3
e. Security is defined at the following levels:									IV. I. 8. 5
(1) File								S	IV. I. 8. 5. 1
(2) Entry								S	IV. I. 8. 5. 2
(3) Group								S	IV. I. 8. 5. 3
(4) Item								S	IV. I. 8. 5. 4
9. Data manipulation language statements	H	H	H	H	H		S		IV. I. 9
a. Control statements (no data movement is accomplished)				H	H		S	S	IV. I. 9. 1
b. Open statement (begin processing of data file):				H	H		S		IV. I. 9. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(1) Mandatory for referencing data								S	IV. I. 9. 2. 1
(2) Identification of part of data base to be processed is required								S	IV. I. 9. 2. 2
(3) Identification of processing mode is required								S	IV. I. 9. 2. 3
(4) Different restrictions apply to opening of sequential and direct access devices								S	IV. I. 9. 2. 4
c. Close statement (finish processing data file):				H	H		S		IV. I. 9. 3
(1) Open and close statements are required in pairs								S	IV. I. 9. 3. 1
(2) Different restrictions apply to opening of sequential and direct access devices								S	IV. I. 9. 3. 2
d. Conditional statements (modify sequence in which host language statements are executed)				H	H		S	S	IV. I. 9. 4
e. Data retrieval statements (data from data base is moved to user working area with no change in data base):				H	H		S	S	IV. I. 9. 5
(1) Locate and access statements:								S	IV. I. 9. 5. 1
(a) Separate statements required for locate and access								S	IV. I. 9. 5. 1. 1
(b) Several retrieval options are available based on:									IV. I. 9. 5. 1. 2
(1 Use of selection criteria								S	IV. I. 9. 5. 1. 2. 1
(2 Data level accessed								S	IV. I. 9. 5. 1. 2. 2
(3 Random or sequential access								S	IV. I. 9. 5. 1. 2. 3
(2) Simple access statements (used to make data available in user working area after location is determined, no selection criteria is used)	H	H	H	H	H		S	S	IV. I. 9. 5. 2

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(a) Statements unnecessary (system already provides general purpose locate and access statement)							S	S	IV. I. 9. 5. 2. 1
(3) Hold or reprocess statements (retain data in user working area for future processing or to lock out access by another run before present run is finished)	H	H	H	H	H		S	S	IV. I. 9. 5. 3
(4) Currency reset statements (reset currency pointers)	H	H	H	H	H		S	S	IV. I. 9. 5. 4
f. Data modification statements:	H	H	H	H	H		S		IV. I. 9. 6
(1) Add:	H	H	H	H	H		S		IV. I. 9. 6. 1
(a) Add data to end of file								S	IV. I. 9. 6. 1. 1
(b) Insert data into a file								S	IV. I. 9. 6. 1. 2
(c) Populate a null file								S	IV. I. 9. 6. 1. 3
(d) Different statements used to perform different add capabilities								S	IV. I. 9. 6. 1. 4
(2) Change (change item values within instance of entries and groups which previously existed within the data base)	H	H	H	H	H		S	S	IV. I. 9. 6. 2
(3) Delete:	H	H	H	H	H		S		IV. I. 9. 6. 3
(a) Group only								S	IV. I. 9. 6. 3. 1
(b) Group and all dependent groups								S	IV. I. 9. 6. 3. 2
(c) Group and optional dependent groups . .								S	IV. I. 9. 6. 3. 3
(d) Different statements used to perform different delete capabilities								S	IV. I. 9. 6. 3. 4
(4) Reorder statements (reorder the sequence of entries in a file or group within a logical string, e.g., sort	H	H	H	H	H		S	S	IV. I. 9. 6. 4

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO.
g. Special purpose statements (handling data in a communications environment or in primary storage):	H	H	H	H	H		S		IV. I. 9. 7
(1) Table handling capability								S	IV. I. 9. 7. 1
(2) Communications (transfer groups of items, called messages or transactions between user working area and queues often associated with external terminal devices):								S	IV. I. 9. 7. 2
(a) Incoming transactions may be used to interrogate or update the data base							S	S	IV. I. 9. 7. 2. 1
(b) Transfer of transactions is accomplished using the same statement provided for data base manipulation							S	S	IV. I. 9. 7. 2. 2
B. Hardware Environment		H	H						IV. II
1. Minimum hardware configuration									IV. II. 1
a. Processors						S		S	IV. II. 1. 1
b. Minimum memory:									IV. II. 1. 2
(1) Batch						S		S	IV. II. 1. 2. 1
(2) On-line						S		S	IV. II. 1. 2. 2
c. Required hardware options, e.g., decimal, arithmetic, real time clock, drum storage, etc.						S		S	IV. II. 1. 3
2. Data base storage devices:									IV. II. 2
a. Required storage devices:									IV. II. 2. 1
(1) Operating system only:									IV. II. 2. 1. 1

(Columns A–H are under the heading TECHNIQUE.)

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(a) Tape						S		S	IV. II. 2. 1. 1. 1
(b) Direct access devices						S		S	IV. II. 2. 1. 1. 2
(2) Addition for minimum data base system:									IV. II. 2. 1. 2
(a) Tape						S		S	IV. II. 2. 1. 2. 1
(b) Direct access devices						S		S	IV. II. 2. 1. 2. 2
(3) Supported media for data base:									IV. II. 2. 1. 3
(a) Random access devices only								S	IV. II. 2. 1. 3. 1
(b) Any device supported by the particular operating system								S	IV. II. 2. 1. 3. 2
(c) Tape								S	IV. II. 2. 1. 3. 3
3. Terminal equipment									IV. II. 3
a. Traffic volume:									IV. II. 3. 1
(1) Maximum number of on-line consoles or terminals that can be connected to the system						S		S	IV. II. 3. 1. 1
(2) Maximum number of consoles that may be active at a given time						S		S	IV. II. 3. 1. 2
(3) Maximum number of on-line users who may have jobs being processed						S		S	IV. II. 3. 1. 3
b. Machine interface						S		S	IV. II. 3. 2
c. System start-up procedure:									IV. II. 3. 3
(1) Manual (data phone to remote site, verbal to computer operator)						S		S	IV. II. 3. 3. 1
(2) Automatic (interrupt compiler from on-line console)						S		S	IV. II. 3. 3. 2
d. System sign-off procedure:									IV. II. 3. 4

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(1) Manual							S	S	IV. II. 3. 4. 1
(2) Automatic (sign off signal to system)							S	S	IV. II. 3. 4. 2
(3) Console input message to remote computer cperator							S	S	IV. II. 3. 4. 3
e. Equipment:									IV. II. 3. 5
(1) CRT:									IV. II. 3. 5. 1
(a) Dark room required								S	IV. II. 3. 5. 1. 1
(b) Output presentation:									IV. II. 3. 5. 1. 2
(1 En masse							S	S	IV. II. 3. 5. 1. 2. 1
(2 Line-by-line							S	S	IV. II. 3. 5. 1. 2. 2
(c) Cursor:									IV. II. 3. 5. 1. 3
(1 Destructive							S	S	IV. II. 3. 5. 1. 3. 1
(2 Non-destructive							S	S	IV. II. 3. 5. 1. 3. 2
(3 Cursor can be positioned anywhere on the screen							S	S	IV. II. 3. 5. 1. 3. 3
(4 Any display (contents of CRT screen) can be printed by on-line user:							S	S	IV. II. 3. 5. 1. 3. 4
-a- Typed command (software)								S	IV. II. 3. 5. 1. 3. 4. 1
-1- Remote printer							S	S	IV. II. 3. 5. 1. 3. 4. 1. 1
-2- Available printer							S	S	IV. II. 3. 5. 1. 3. 4. 1. 2
-b- Special purpose key (hardware)								S	IV. II. 3. 5. 1. 3. 4. 2
-1- Remote printer							S	S	IV. II. 3. 5. 1. 3. 4. 2. 1
-2- Available printer							S	S	IV. II. 3. 5. 1. 3. 4. 2. 2
-c- Size of display:								S	IV. II. 3. 5. 1. 3. 4. 3

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
-1- Number of lines						S		S	IV. II. 3. 5. 1. 3. 4. 3. 1
-2- Characters per line						S		S	IV. II. 3. 5. 1. 3. 4. 3. 2
-d- Available character set						S		S	IV. II. 3. 5. 1. 3. 4. 4
(2) Teletype:									IV. II. 3. 5. 2
(a) Characters per second						S		S	IV. II. 3. 5. 2. 1
(b) Typing errors corrected by:							S		IV. II. 3. 5. 2. 2
(1 Backspace							S	S	IV. II. 3. 5. 2. 2. 1
(2 Erase character							S	S	IV. II. 3. 5. 2. 2. 2
(3 Delete line/command/query								S	IV. II. 3. 5. 2. 2. 3
(c) Noise level:									IV. II. 3. 5. 2. 3
(1 Negligible							S	S	IV. II. 3. 5. 2. 3. 1
(2 Otherwise							S	S	IV. II. 3. 5. 2. 3. 2
(3) Keyboard:									IV. II. 3. 5. 3
(a) System reserved keyboard characters not available to user						S	S	S	IV. II. 3. 5. 3. 1
(b) Number of special command keys						S		S	IV. II. 3. 5. 3. 2
4. Recovery procedure for type or format error:									IV. II. 4
a. Correct a character							S	S	IV. II. 4. 1
b. Correct a line							S	S	IV. II. 4. 2
c. Entire procedure must be reentered							S	S	IV. II. 4. 3
C. Operating System		H	H		H				IV. III
1. Operating Environment					H				IV. III. 1
a. Uniprogramming system							S	S	IV. III. 1. 1

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
	TECHNIQUE								
b. Multiprogramming system:							S	S	IV. III. 1. 2
(1) Number of co-resident jobs							S	S	IV. III. 1. 2. 1
(2) Number of simultaneous jobs							S	S	IV. III. 1. 2. 2
(3) Dismissal of a program when it is completed, blocked or has exhausted its time quantum							S	S	IV. III. 1. 2. 3
(4) Roll-in/roll-out techniques (more than one program physically in main memory)							S	S	IV. III. 1. 2. 4
c. Multiprocessing							S	S	IV. III. 1. 3
2. Scheduling									IV. III. 2
a. Data base integrity (protection from programs external to system):									IV. III. 2. 1
(1) Operating system used to stop non-system accessing							S	S	IV. III. 2. 1. 1
b. Program scheduling and interrupt handling:									IV. III. 2. 2
(1) Operating system function used							S	S	IV. III. 2. 2. 1
(2) Special scheduling within the DMS							S	S	IV. III. 2. 2. 2
c. Concurrency of operations									IV. III. 2. 3
(1) Concurrent operations on a data base may occur because:									IV. III. 2. 3. 1
(a) DMS allows more than one user to call simultaneously on the same or different functions					H		S	S	IV. III. 2. 3. 1. 1
(b) Operating system allows more than one user to interact with the same copy of the DMS					H		S	S	IV. III. 2. 3. 1. 2

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
(c) Operating system allows more than one copy of the DMS					H		S	S	IV. III. 2. 3. 1. 3
(2) Concurrency during file creation:									IV. III. 2. 3. 2
(a) Two files can be created simultaneously .					H		S	S	IV. III. 2. 3. 2. 1
(b) The creation process on one file can be achieved at the same time as other functions on another file					H		S	S	IV. III. 2. 3. 2. 2
(3) Concurrency with single copy (when only one copy of DMS exists)									IV. III. 2. 3. 3
(a) One application program called by two users who interact with different data files					H		S	S	IV. III. 2. 3. 3. 1
(b) One application program interacting with the same data file, called concurrently by two users					H		S	S	IV. III. 2. 3. 3. 2
(c) More than one application program interacting with different data files					H		S	S	IV. III. 2. 3. 3. 3
(4) Concurrency with multiple copies (when more than one copy of the DMS is allowed within the operating system):									IV. III. 2. 3. 4
(a) Concurrent operations limited to one function, e.g., multiple concurrent querying of a file, but no concurrency of update					H		S	S	IV. III. 2. 3. 4. 1
(b) Concurrent operations can only be performed on different data files					H		S	S	IV. III. 2. 3. 4. 2
(c) Concurrent operations can be performed on both the same and different data files .					H		S	S	IV. III. 2. 3. 4. 3
3. Software Facilities	H	H	H		H		S		IV. III. 3

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
a. Use of operating system facilities by the DMS . .									IV. III. 3. 1
(1) Access methods (the basic I/O facilities of the operating system)	H		H	H	H		S	S	IV. III. 3. 1. 1
(2) Task scheduler	H	H	H	H	H		S	S	IV. III. 3. 1. 2
(3) Loader link editor	H	H	H	H	H		S	S	IV. III. 3. 1. 3
(4) Space and resource management (includes allocation of buffers and opening and closing of files)	H			H	H		S	S	IV. III. 3. 1. 4
(5) Communication facilities	H	H	H	H	H		S	S	IV. III. 3. 1. 5
(6) Additional features (sort programs, library utilities)	H			H	H	S	S	S	IV. III. 3. 1. 6
(7) Communications subsystems (terminal input and output control)	H	H	H	H	H		S	S	IV. III. 3. 1. 7
(8) Polling and queueing of messages is done by operating system	H			H	H		S	S	IV. III. 3. 1. 8
(9) Major scheduling of users is the responsibility of the DMS	H				H		S	S	IV. III. 3. 1. 9
b. Other software:									IV. III. 3. 2
(1) Compiler							S	S	IV. III. 3. 2. 1
(2) Sort/merge							S	S	IV. III. 3. 2. 2
4. Modes of System Use									IV. III. 4
a. Interactive mode		H	H				S		IV. III. 4. 1
(1) Prestored procedures (system does not actively assist the user):									IV. III. 4. 1. 1
(a) Execution required directly from terminal							S	S	IV. III. 4. 1. 1. 1
(b) All necessary parameters supplied from terminal							S	S	IV. III. 4. 1. 1. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
(c) Capability is available to which DMS functions						S	S	S	IV. III. 4. 1. 1. 3
b. Conversational mode		H	H				S		IV. III. 4. 2
(1) Capability is available to which DMS functions						S	S	S	IV. III. 4. 2. 1
(2) Scenario driven (walk thru)							S	S	IV. III. 4. 2. 2
(3) User can request tutorial assistance at any time							S	S	IV. III. 4. 2. 3
(4) User can suppress tutorial assistance at any time							S	S	IV. III. 4. 2. 4
(5) Tutorials can be added							S	S	IV. III. 4. 2. 5
(6) Tutorials can be deleted							S	S	IV. III. 4. 2. 6
(7) Tutorials can be modified							S	S	IV. III. 4. 2. 7
(8) User can change from one level of dialogue to another at any time							S	S	IV. III. 4. 2. 8
(9) Specific tutorials provided for:									IV. III. 4. 2. 9
(a) File definition								S	IV. III. 4. 2. 9. 1
(b) File creation								S	IV. III. 4. 2. 9. 2
(c) File restructuring								S	IV. III. 4. 2. 9. 3
(d) Language use								S	IV. III. 4. 2. 9. 4
(e) Query formulation								S	IV. III. 4. 2. 9. 5
(f) Output processing								S	IV. III. 4. 2. 9. 6
(g) Error procedures								S	IV. III. 4. 2. 9. 7
(h) Recovery procedures								S	IV. III. 4. 2. 9. 8
(10) Acknowledgement:									IV. III. 4. 2. 10
(a) Acknowledgement of all user-initiated dialogue provided optionally at the user's request							S	S	IV. III. 4. 2. 10. 1
5. Batched Processing	H	H	H	H	H		S		IV. III. 5

TECHNIQUE spans columns A–H.

ATTRIBUTES		TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
a.	Multiple tasks									IV. III. 5. 1
	(1) Accumulation against a single file							S	S	IV. III. 5. 1. 1
	(2) Accumulation against a given collection of files							S	S	IV. III. 5. 1. 2
	(3) Processing of accumulated multiple tasks may be done in batch mode							S	S	IV. III. 5. 1. 3
b.	Jobs may be entered through a remote terminal for:							S		IV. III. 5. 2
	(1) File generation							S	S	IV. III. 5. 2. 1
	(2) Maintenance							S	S	IV. III. 5. 2. 2
	(3) Retrieval							S	S	IV. III. 5. 2. 3
	(4) Output							S	S	IV. III. 5. 2. 4
c.	Jobs may be entered through a local terminal for:									IV. III. 5. 3
	(1) File generation							S	S	IV. III. 5. 3. 1
	(2) Maintenance							S	S	IV. III. 5. 3. 2
	(3) Retrieval							S	S	IV. III. 5. 3. 3
	(4) Output							S	S	IV. III. 5. 3. 4
d.	Composition of job request at a remote terminal							S	S	IV. III. 5. 4
e.	Composition of job request at a local terminal							S	S	IV. III. 5. 5
f.	Request library procedures from a remote terminal.							S	S	IV. III. 5. 6
g.	Request library procedures from a local terminal.							S	S	IV. III. 5. 7
h.	Parameters can be entered for a library procedure from a local terminal							S	S	IV. III. 5. 8
i.	Library procedures may be modified prior to execution:									IV. III. 5. 9
	(1) Temporarily							S	S	IV. III. 5. 9. 1
	(2) Permanently							S	S	IV. III. 5. 9. 2

ATTRIBUTES	A	B	C	D	E	F	G	H	REFERENCE NO
j. System informs user when:									IV. III. 5. 10
(1) Job is accepted							S	S	IV. III. 5. 10. 1
(2) Job errors are encountered							S	S	IV. III. 5. 10. 2
(3) Job is terminated							S	S	IV. III. 5. 10. 3
k. Maximum number of active users that can operate simultaneously from remote terminals						S	S	S	IV. III. 5. 11
6. DMS Transferability									IV. III. 6
a. System can be transferred between computers of the same family operating under the same and different versions of the operating system, or even in an entirely different operating system		H	H				S	S	IV. III. 6. 1
b. System can be transferred to operate on computers that are not in the same family		H	H				S	S	IV. III. 6. 2
c. User must modify his procedures when transferring to different computers of the same family or a different computer family		H	H				S	S	IV. III. 6. 3
d. User must transfer his data base from one type of storage device to another, either on the same or different computers or translate his data from one representation to another		H	H				S	S	IV. III. 6. 4
D. Documentation Availability									IV. IV
1. System specifications									IV. IV. 1
a. System description (overview)						S		S	IV. IV. 1. 1
b. Conceptual system description						S		S	IV. IV. 1. 2
c. Detailed design specification						S		S	IV. IV. 1. 3
d. Implementation specification						S		S	IV. IV. 1. 4
2. Operation documentation									IV. IV. 2

ATTRIBUTES	TECHNIQUE A	B	C	D	E	F	G	H	REFERENCE NO
a. User:									IV. IV. 2. 1
(1) File design guide						S		S	IV. IV. 2. 1. 1
(2) Language reference manual						S		S	IV. IV. 2. 1. 2
(3) System analysis guide						S		S	IV. IV. 2. 1. 3
b. System operation:									IV. IV. 2. 2
(1) Operator's manual						S		S	IV. IV. 2. 2. 1
(2) System maintenance						S		S	IV. IV. 2. 2. 2
(3) System administrative procedures						S		S	IV. IV. 2. 2. 3

4. Measurement/Test Pairs

This segment uses the results of the linking process in Part 3 and tells how the test technique will be used to evaluate and test the DMS characteristics. This process is shown in a list form. Each test technique is shown, followed by the reference numbers of the characteristics that can be tested by the technique. Following each reference number is the measurement(s) that will be used to evaluate/test the characteristics utilizing the given test technique.

MEASUREMENT/TEST PAIRS

Test Technique: Benchmark Tests

Reference Number	Measurement
I. II. 3	Gross timings.
I. II. 3. 1	Gross timings.
I. II. 3. 2	Gross timings.
I. II. 3. 3	Gross timings.
I. II. 3. 4	Gross timings.
I. II. 3. 5	Gross timings.
I. II. 3. 6. 1	Gross timings.
I. II. 3. 6. 2	Gross timings.
I. II. 3. 6. 3	Gross timings.
I. II. 3. 6. 4	Gross timings.
I. II. 3. 6. 5	Gross timings.
I. II. 3. 7	Gross timings.
I. II. 3. 8	Gross timings.
I. II. 4	Gross timings.
I. II. 4. 1	Gross timings.
I. II. 4. 2	Gross timings.
I. II. 4. 3	Gross timings.
I. II. 4. 4	Gross timings.
I. II. 4. 5	Gross timings.
I. II. 4. 6	Gross timings.
I. II. 4. 7	Gross timings.
I. III. 1. 2. 2. 1	Gross timings.
I. III. 1. 2. 2. 2	Gross timings.
I. III. 2. 1	Gross timings.
I. III. 2. 2	Gross timings.
I. III. 2. 2. 1	Gross timings.
I. III. 3. 1	Gross timing-compare to OS access method timings.
I. III. 3. 2. 2. 1	Gross timing-compare to timings of I. III. 3. 2. 2. 2.
I. III. 3. 2. 2. 2	Gross timing-compare to timings of I. III. 3. 2. 2. 1.
I. III. 3. 2. 2. 3	Gross timings.
I. III. 3. 2. 2. 4. 1	Gross timings.
I. III. 3. 2. 2. 4. 2	Gross timings.
I. III. 3. 2. 2. 4. 3	Gross timings.
I. III. 4	Gross timings.
I. III. 4. 1	Gross timings.
I. III. 4. 1. 2	Gross timings.
I. III. 4. 2	Gross timings.
I. III. 4. 2. 1	Gross timings.
I. III. 4. 2. 3	Gross timings.
I. III. 4. 2. 4	Gross timings.

Reference Number	Measurement
II	Gross timings.
II. I	Gross timings.
II. I. 2	Gross timings.
II. I. 2. 1	Gross timings.
II. I. 4	Gross timings.
II. II	Gross timings.
II. II. 2. 4. 3	Gross timings.
II. II. 4	Gross timings.
II. II. 5	Gross timings.
II. II. 5. 1	Gross timings.
II. II. 5. 2	Gross timings.
II. II. 5. 3	Gross timings.
II. II. 5. 3. 2	Gross timings.
II. II. 5. 3. 3	Gross timings.
II. II. 5. 3. 3. 2	Gross timings.
II. II. 5. 3. 3. 3	Gross timings.
II. II. 5. 3. 3. 4	Gross timings.
II. III	Gross timings.
II. III. 1	Gross timings.
II. III. 2	Gross timings.
II. III. 3	Gross timings.
II. III. 4	Gross timings.
II. III. 5	Gross timings.
II. III. 6	Gross timings.
II. III. 6. 1	Gross timings.
II. III. 6. 2	Gross timings.
III. I. 2. 1. 2. 1	Gross timings.
III. I. 2. 1. 2. 2	Gross timings.
III. I. 2. 1. 2. 3	Gross timings.
IV. I. 3	Gross timings.
IV. I. 3. 1	Gross timings.
IV. I. 3. 2	Gross timings.
IV. I. 3. 3	Gross timings.
IV. I. 9	Gross timings.
IV. I. 9. 5. 2	Gross timings.
IV. I. 9. 5. 3	Gross timings.
IV. I. 9. 5. 4	Gross timings.
IV. I. 9. 6	Gross timings.
IV. I. 9. 6. 1	Gross timings.
IV. I. 9. 6. 2	Gross timings.
IV. I. 9. 6. 3	Gross timings.
IV. I. 9. 6. 4	Gross timings.
IV. I. 9. 7	Gross timings.

Reference Number	Measurement
IV. III. 3	Gross timings.
IV. III. 3. 1. 1	Gross timings.
IV. III. 3. 1. 2	Gross timings.
IV. III. 3. 1. 3	Gross timings.
IV. III. 3. 1. 4	Gross timings.
IV. III. 3. 1. 5	Gross timings.
IV. III. 3. 1. 6	Gross timings.
IV. III. 3. 1. 7	Gross timings.
IV. III. 3. 1. 8	Gross timings.
IV. III. 3. 1. 9	Gross timings.
IV. III. 5	Gross timings.

MEASUREMENT/TEST PAIRS

Test Technique: Modeling/Simulation

Reference Number	Measurement
I. I	Obtain operational timings on varying logical structures.
I. III. 1. 2. 1. 1	I/O timings.
I. III. 1. 2. 1. 2	I/O timings.
I. III. 1. 2. 2. 1	Access timings.
I. III. 1. 2. 2. 2	Access timings.
I. III. 2. 1	Access timings.
I. III. 2. 2	Access timings.
I. III. 2. 2. 1	Access timings.
I. III. 3. 1	Timing differences, overhead differences between OS and DMS methods.
I. III. 3. 2. 2	Overhead figures for indexing.
I. III. 3. 2. 2. 1	Overhead figures for chaining.
I. III. 3. 2. 2. 2	Overhead figures for chaining.
I. III. 3. 2. 2. 3	Overhead figures for chaining.
I. III. 3. 2. 2. 4	Overhead figures for chaining.
I. III. 3. 2. 2. 4. 1	Overhead figures for chaining.
I. III. 3. 2. 2. 4. 2	Overhead figures for chaining.
I. III. 3. 2. 2. 4. 3	Overhead figures for chaining.
I. III. 4	Access timings, average number of seeks per record.
I. III. 4. 1	Overhead figures for indexing, access timings.
I. III. 4. 2	Overhead figures concerned with randomizing, access timings.
I. III. 4. 2. 1	Access timings,
I. III. 4. 2. 3	Access timings, overhead figures.
I. III. 4. 2. 4	Access timings, overhead figures.
II	I/O timings, device usage statistics, channel activity, overhead figures.
II. I	I/O timings, device usage statistics, channel activity, overhead figures.
II. I. 2	I/O overlap, device usage statistics, overhead figures.
II. I. 2. 1	I/O overlap, device usage statistics, overhead figures.
II. I. 4	I/O timings, device usage statistics, gross timings.
II. I. 5	Overhead figures, device usage statistics.
II. II	Overhead figures, device usage statistics, channel activity and overlap.
II. II. 4	Overhead figures, device usage statistics.
II. II. 5	I/O activity figures, device usage statistics, access timings, overhead figures.

Reference Number	Measurement
II. II. 5. 1	Access timings, average number of seeks, I/O timings and overlap.
II. II. 5. 2	Access timings, I/O timings, device usage statistics.
II. II. 5. 3	Overhead figures, I/O timings, device usage statistics, gross timings.
II. II. 5. 3. 2	Overhead figures, I/O timings, device usage statistics, gross timings.
II. II. 5. 3. 3	Overhead figures, I/O timings, device usage statistics, gross timings.
II. II. 5. 3. 3. 2	Overhead figures, I/O timings, device usage statistics, gross timings.
II. II. 5. 3. 3. 3	Overhead figures, I/O timings, device usage statistics, gross timings.
II. II. 5. 3. 3. 4	Overhead figures, I/O timings, device usage statistics, gross timings.
II. II. 6	Overhead figures, I/O usage statistics.
II. III	Access timings, device usage statistics, I/O file activity.
II. III. 2	Access timings, device usage statistics, I/O file activity, I/O overlap.
II. III. 3	Device usage statistics, I/O activity and overlap, overhead figures.
II. III. 4	Device usage statistics, I/O activity and overlap, overhead figures.
II. III. 5	Device usage statistics, I/O activity and overlap, overhead figures.
II. III. 6	Device usage statistics, I/O activity and overlap, overhead figures.
II. III. 6. 1	Device usage statistics, I/O activity and overlap, overhead figures.
II. III. 6. 2	Device usage statistics, I/O activity and overlap, overhead figures.
II. III. 6. 6	Device usage statistics, channel activity, I/O overlap.
II. III. 6. 6. 3	Device usage statistics, channel activity, I/O overlap, overhead figures.
II. III. 7	Device usage statistics, channel activity, I/O overlap.
II. III. 8	Device usage statistics, channel activity, I/O overlap.
II. IV. 1	Module activity, gross timings.
II. IV. 2	Module activity, gross timings.
II. IV. 3	Module activity, gross timings.
II. IV. 4	Module activity, gross timings.
II. IV. 5	Module activity, gross timings.
III	I/O activity, overhead figures, device usage statistics, module activity.

Reference Number	Measurement
III. II	I/O activity, overhead figures, device usage statistics.
III. III	I/O activity, overhead figures, device usage statistics.
III. IV	I/O activity, overhead figures, device usage statistics.
IV	I/O activity and overlap, overhead figures, device usage statistics, module activity.
IV. I. 2	Channel activity, I/O timings, overhead figures, access timings.
IV. I. 3	Access timings, average number of seeks per hit, I/O overlap and timings.
IV. I. 3. 1	Access timings, average number of seeks per hit, I/O overlap and timings.
IV. I. 3. 2	Access timings, average number of seeks per hit, I/O overlap and timings.
IV. I. 3. 3	Access timings, average number of seeks per hit, I/O overlap and timings.
IV. I. 7	Overhead figures, I/O usage and overlap.
IV. I. 9	Module activity, gross timings.
IV. I. 9. 5. 2	Module activity, gross timings.
IV. I. 9. 5. 3	Module activity, gross timings.
IV. I. 9. 5. 4	Module activity, gross timings.
IV. I. 9. 6	Module activity, gross timings.
IV. I. 9. 6. 1	Module activity, gross timings.
IV. I. 9. 6. 2	Module activity, gross timings.
IV. I. 9. 6. 3	Module activity, gross timings.
IV. I. 9. 6. 4	Module activity, gross timings.
IV. I. 9. 7	Module activity, gross timings.
IV. II	Device usage statistics, I/O channel activity and overlap, I/O timings.
IV. III	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 3	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 3. 1. 2	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 3. 1. 3	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 3. 1. 5	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 3. 1. 7	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.

Reference Number	Measurement
IV. III. 4. 1	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 4. 2	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 5	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 6. 1	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 6. 2	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 6. 3	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.
IV. III. 6. 4	Module activity, device usage statistics, I/O activity-overlap, overhead figures, scheduling activity, allocation.

MEASUREMENT/TEST PAIRS

Test Technique: Hardware Monitors

Reference Number	Measurement
I. III	Channel activity, I/O activity, gross timings.
I. III. 1. 2. 2. 1	Core usage statistics, channel-I/O activity and timings.
I. III. 1. 2. 2. 2	Core usage statistics, channel-I/O activity and timings.
I. III. 4	Channel activity, I/O activity and timings, access timings.
I. III. 4. 1	Channel activity, I/O activity and timings, access timings, core allocation.
I. III. 4. 2	Channel activity, I/O activity and timings, access timings, core allocation.
II	Channel activity, I/O activity and timings, core allocation.
II. I	Channel activity, I/O activity and timings, core allocation.
II. II	Channel activity, I/O activity and timings, core allocation.
II. II. 5	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. II. 5. 1	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. II. 5. 1. 1. 2	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. II. 5. 1. 1. 3	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. II. 5. 1. 2. 1	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. II. 5. 1. 2. 2	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. II. 5. 2	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. III	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. III. 2	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. III. 3	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. III. 4	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. III. 6. 6	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. III. 6. 6. 1	Channel activity, I/O activity and timings, core allocation, I/O overlap.
II. III. 6. 6. 2	Channel activity, I/O activity and timings, core allocation, I/O overlap.

Reference Number	Measurement
III	Channel activity, device usage statistics, I/O timings and overlap
III. I	Channel activity, device usage statistics, I/O timings and overlap
III. I. 1. 3. 1	CPU timings.
III. I. 1. 3. 2	CPU timings.
III. I. 1. 3. 3	CPU timings.
III. I. 1. 3. 4	CPU timings.
III. I. 1. 3. 5	Module activity.
III. I. 2. 1. 1. 2	Core usage statistics.
III. I. 2. 1. 1. 3	Device usage statistics, I/O activity.
III. I. 2. 1. 2. 1	CPU timings, core/module usage statistics.
III. I. 2. 1. 2. 2	CPU timings, core/module usage statistics.
III. I. 2. 1. 2. 3	CPU timings, core/module usage statistics.
III. I. 2. 1. 3. 2. 4	I/O timings, module usage statistics.
III. I. 2. 1. 3. 2. 5	CPU timings, module activity statistics.
III. I. 2. 1. 3. 2. 7	Module activity statistics.
III. I. 2. 1. 3. 2. 8	Module activity statistics, I/O usage and timings.
III. I. 3. 2. 1. 2	Module activity statistics, CPU time.
III. I. 3. 2. 1. 3	Module activity statistics, CPU time.
III. I. 4. 2. 1. 1. 4	Core usage statistics.
III. I. 4. 2. 1. 1. 5	Channel activity statistics, I/O activity statistics.
III. I. 4. 2. 2. 1. 1	Module activity statistics.
III. I. 4. 2. 2. 1. 2	Module activity statistics.
III. I. 4. 2. 2. 1. 3	Module activity statistics.
III. II	Module activity statistics.
III. II. 2	Module activity statistics.
III. II. 4. 5. 1	Module activity statistics.
III. II. 4. 5. 4	Module activity statistics.
III. II. 4. 7. 2. 1	CPU timings.
III. II. 4. 7. 2. 2	CPU timings.
III. II. 4. 7. 2. 3	CPU timings.
III. II. 4. 9. 4. 1	Module activity statistics.
III. II. 4. 9. 4. 2	CPU timings.
III. III. 1. 2	Module activity statistics.
III. III. 1. 3	Module activity statistics.
III. III. 1. 4	Module activity statistics.
IV	Channel activity, I/O activity and overlaps, module activity.
IV. I. 3	Channel activity, I/O activity and overlaps, timings.
IV. I. 3. 1	Channel activity, I/O activity and overlaps, timings.
IV. I. 3. 2	Channel activity, I/O activity and overlaps, timings.
IV. I. 3. 3	Channel activity, I/O activity and overlaps, timings.
IV. I. 9	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 1	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 2	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 3	Channel activity, I/O activity and overlaps, timings.

Reference Number	Measurement
IV. I. 9. 4	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 5	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 5. 2	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 5. 3	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 5. 4	Channel activity, I/O activity and overlaps timings.
IV. I. 9. 6	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 6. 1	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 6. 2	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 6. 3	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 6. 4	Channel activity, I/O activity and overlaps, timings.
IV. I. 9. 7	Channel activity, I/O activity and overlaps, timings.
IV. III. 3. 1. 1	Core usage, I/O and channel activity, timings, module activity.
IV. III. 3. 1. 2	Core usage, I/O and channel activity, timings, module activity.
IV. III. 3. 1. 3	Core usage, I/O and channel activity, timings, module activity.
IV. III. 3. 1. 4	Core usage, I/O and channel activity, timings, module activity.
IV. III. 3. 1. 5	Core usage, I/O and channel activity, timings, module activity.
IV. III. 3. 1. 6	Core usage, I/O and channel activity, timings, module activity.
IV. III. 3. 1. 7	Core usage, I/O and channel activity, timings, module activity.
IV. III. 3. 1. 8	Core usage, I/O and channel activity, timings, module activity.
IV. III. 5	Core usage, I/O and channel activity, timings, module activity.

MEASUREMENT/TEST PAIRS

Test Technique: Software Monitors

Reference Number	Measurement
I. II. 3	Gross timings, overhead figures, module activities.
I. II. 4	Gross timings, overhead figures, module activities.
I. II. 4. 2	Gross timings, overhead figures, module activities.
I. II. 4. 3	Gross timings, overhead figures, module activities.
I. II. 4. 4	Gross timings, overhead figures, module activities.
I. II. 4. 5	Gross timings, overhead figures, module activities.
I. II. 4. 6	Gross timings, overhead figures, module activities.
I. II. 4. 7	Gross timings, overhead figures, module activities.
I. III	Gross timings, overhead figures, module activities, I/O activity/overlap.
I. III. 1. 2. 2. 1	Gross timings, overhead figures, access timings, I/O activity/overlap.
I. III. 1. 2. 2. 2	Same as I. III. 1. 2. 2. 1-compare.
I. III. 2. 1	Average seek time per record.
I. III. 2. 2	Same as I. III. 2. 1-compare.
I. III. 2. 2. 1	Same as I. III. 2. 1-2. 2-compare.
I. III. 3. 1	Same as I. III. 1. 2. 2. 1 and 1. 2. 2. 2.
I. III. 3. 2. 2	Gross average record access time, overhead figures.
I. III. 3. 2. 2. 1	Gross average record access time, overhead figures.
I. III. 3. 2. 2. 2	Gross average record access time, overhead figures.
I. III. 3. 2. 2. 3	Gross average record access time, overhead figures.
I. III. 3. 2. 2. 4. 1	Overhead figures.
I. III. 3. 2. 2. 4. 2	Overhead figures.
I. III. 3. 2. 2. 4. 3	Overhead figures.
I. III. 3. 2. 2. 5	Gross average record access time, overhead figures.
I. III. 4	Average access times, overhead figures, module activity.
I. III. 4. 1	Average access times, overhead figures.
I. III. 4. 1. 2. 1	Average record access timings, overhead figures. } compare
I. III. 4. 1. 2. 2	Average record access timings, overhead figures. } compare
I. III. 4. 1. 2. 3	Average record access timings, overhead figures. } compare
I. III. 4. 2	Average record access timings, overhead figures, module activity.
I. III. 4. 2. 1	Average record access timings, overhead figures.
I. III. 4. 2. 2	Average record access timings, overhead figures.
I. III. 4. 2. 2. 1	Average record access timings, overhead figures.
I. III. 4. 2. 3	Average record access timings, overhead figures.
I. III. 4. 2. 4	Average record access timings, overhead figures.
II	Gross timings, channel activity, I/O overlap, module activity.

Reference Number	Measurement
II. I. 4	Gross timings, channel activity, I/O overlap, module activity.
II. I. 5	Module activity, timings, overhead figures.
II. II	Module activity, timings, overhead figures, channel activity, I/O overlaps.
II. II. 5	Module activity, timings, overhead figures, channel activity, I/O overlaps, access time.
II. II. 5. 1	Module activity, timings, overhead figures, channel activity, I/O overlaps, access time.
II. II. 5. 1. 1. 2	Channel activity, I/O overlap.
II. II. 5. 1. 1. 3	Channel activity, I/O overlap.
II. II. 5. 2	Same as II. II. 5.
II. II. 5. 3	Same as II. II. 5
II. II. 6	Overhead figures.
II. III	Gross timings, module activity.
II. III. 2	Gross timings, module activity, average access time, I/O activities.
II. III. 3	Gross timings, module activity, average access time, I/O activities.
II. III. 4	Gross timings, module activity, average access time, I/O activities, overhead.
II. III. 5	Gross timings, module activity, average access time, I/O activities.
II. III. 6	Gross timings, module activity, average access time, I/O activities.
II. III. 6. 1	Gross timings, module activity, average access time, I/O activities.
II. III. 6. 2	Gross timings, module activity, average access time, I/O activities.
II. III. 7	Gross timings, module activity, average access time, I/O activities.
II. III. 8	Gross timings, module activity, average access time, I/O activities.
III	Module activity, overhead involved, channel and I/O activity, overlap.
III. I	Module activity, overhead involved, channel and I/O activity, overlap.
III. I. 1. 3. 1	Overhead figures.
III. I. 1. 3. 2	Overhead figures.
III. I. 1. 3. 3	Overhead figures.
III. I. 1. 3. 4	Overhead figures.
III. I. 1. 3. 5	Overhead figures.

Reference Number	Measurement
III. I. 2. 1. 1. 1	Overhead figures.
III. I. 2. 1. 1. 2	Overhead figures, module activity.
III. I. 2. 1. 1. 3	Overhead figures, device (channel) usage.
III. I. 2. 1. 1. 4	Overhead figures.
III. I. 2. 1. 2. 1	Overhead figures.
III. I. 2. 1. 2. 2	Overhead figures, module activity.
III. I. 2. 1. 2. 3	Overhead figures, access timings.
III. I. 2. 1. 3. 1	Overhead figures.
III. I. 2. 1. 3. 2. 4	Overhead figures, I/O activities and timings.
III. I. 2. 1. 3. 2. 5	Overhead figures, CPU timings, module activity.
III. I. 2. 1. 3. 2. 6	Overhead figures, module activities.
III. I. 2. 1. 3. 2. 7	Overhead figures, module activities.
III. I. 2. 1. 3. 2. 8	Overhead figures, module activities, access timings.
III. I. 2. 1. 3. 2. 9	Overhead figures.
III. I. 2. 1. 3. 2. 10	Overhead figures.
III. I. 3. 1. 1. 2. 1	Overhead figures, module activity.
III. I. 3. 1. 1. 2. 2	Overhead figures, module activity.
III. I. 3. 1. 1. 3	Overhead figures.
III. I. 3. 2. 1. 1	Overhead figures.
III. I. 4. 2. 1. 1. 1	Module activity.
III. I. 4. 2. 1. 1. 4	Core usage statistics.
III. I. 4. 2. 1. 1. 5	Channel activity.
III. II	Overhead figures, gross timings, module activity.
III. II. 1	Overhead figures, gross timings, module activity.
III. II. 3. 5	Overhead figures.
III. II. 3. 6	Overhead figures.
III. II. 4. 2. 1	Overhead figures.
III. II. 4. 2. 2	Overhead figures.
III. II. 4. 2. 3	Overhead figures.
III. II. 4. 2. 4	Overhead figures.
III. II. 4. 5. 1	Overhead figures.
III. II. 4. 5. 2. 1	Overhead figures.
III. II. 4. 5. 2. 2	Overhead figures.
III. II. 4. 5. 3	Overhead figures.
III. II. 4. 5. 4	Overhead figures.
III. II. 4. 9. 1	Overhead figures, module activity.
III. II. 4. 9. 2	Overhead figures.
III. II. 4. 9. 3	Overhead figures.
III. II. 4. 9. 4. 1	Overhead figures, module activity.
III. II. 4. 9. 4. 2	Overhead figures, CPU timings.
III. IV	Overhead figures, module activity.
IV	Module activity, I/O and channel activities, gross timings, access timings.

Reference Number	Measurements
IV. I. 3	Module activity, I/O and channel activities, gross timings, access timings.
IV. I. 3. 1	Module activity, I/O and channel activities, gross timings, access timings.
IV. I. 3. 2	Module activity, I/O and channel activities, gross timings, access timings.
IV. I. 3. 3	Module activity, I/O and channel activities, gross timings, access timings.
IV. I. 9	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV I. 9. 4	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 5	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 5. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 5. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 5. 4	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 6	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 6. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 6. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 6. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 6. 4	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. I. 9. 7	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 1. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.

Reference Number	Measurements
IV. III. 2. 3. 1. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 1. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 2. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 2. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 3. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 3. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 3. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 4. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 4. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 2. 3. 4. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 1	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 2	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 3	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 4	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 5	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 6	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 7	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 8	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 3. 1. 9	Module activity, gross timings, I/O timings/overlap, overhead figures.
IV. III. 5	Module activity, gross timings, I/O timings/overlap, overhead figures.

MEASUREMENT/TEST PAIRS

Test Technique: Operational Analysis

Reference Number	Measurement
I. I. 1	Operational performance evaluation.
I. II. 3	Operational performance evaluation.
I. II. 4	Operational performance evaluation.
I. II. 4. 2	Operational performance evaluation.
I. II. 4. 3	Operational performance evaluation.
I. II. 4. 4	Operational performance evaluation.
I. II. 4. 5	Operational performance evaluation.
I. II. 4. 6	Operational performance evaluation.
I. II. 4. 7	Operational performance evaluation.
I. III	Overall performance of storage structure.
I. III. 1. 2. 2. 1	Performance/characteristics of the OS-supplied access methods.
I. III. 1. 2. 2. 2	Performance/characteristics of the DMS-supplied access methods.
I. III. 2. 1	Operational performance evaluation.
I. III. 2. 2	Operational performance evaluation.
I. III. 2. 2. 1	Operational performance evaluation.
I. III. 3. 1	Operational performance evaluation.
I. III. 3. 2. 2	Operational performance evaluation.
I. III. 4	Operational performance evaluation.
I. III. 4. 1	Operational performance evaluation.
I. III. 4. 2	Operational performance evaluation.
II	Operational performance evaluation.
II. I. 3. 2. 1	Detect any operational problems of handling files created by this language.
II. I. 3. 2. 2	Detect any operational problems of handling files created by this language.
II. I. 3. 2. 3	Detect any operational problems of handling files created by this language.
II. I. 3. 2. 4	Detect any operational problems of handling files created by this language.
II. I. 3. 2. 5	Find any operational problems with this attribute.
II. I. 3. 3	Find any operational problems with this attribute.
II. I. 3. 4	Find any operational problems with this attribute.
II. I. 4	Detect any problems concerning file population function.
II. II	Detect problems/outstanding points of the Update function.
II. II. 5	Detect operational good and bad points of access and manipulation.

Reference Number	Measurement
II. II. 5. 1	Operational performance analysis.
II. II. 5. 2	Operational performance analysis.
II. II. 5. 3	Operational performance analysis.
II. II. 5. 3. 2	Operational performance analysis.
II. II. 5. 3. 3	Operational performance analysis.
II. II. 5. 3. 3. 2	Operational performance analysis.
II. II. 5. 3. 3. 3	Operational performance analysis.
II. II. 5. 3. 3. 4	Operational performance analysis.
II. II. 6	Operational performance analysis.
II. III	Operational performance analysis.
II. III. 1	Operational performance analysis.
II. III. 2	Operational performance analysis.
II. III. 3	Operational performance analysis.
II. III. 4	Operational performance analysis.
II. III. 5	Operational performance analysis.
II. III. 6	Operational performance analysis.
II. III. 6. 1	Operational performance analysis.
II. III. 6. 2	Operational performance analysis.
II. III. 7	Operational performance analysis.
II. III. 8	Operational performance analysis.
II. IV. 1	Determine ease of use and/or problems with form during operation.
II. IV. 2	Determine if there are enough operands during use plus correctness.
II. IV. 3	Determine if there are enough operators during use plus correctness.
II. IV. 4	Determine reliability and correctness of statistics.
II. IV. 5	Determine reliability and correctness of the conditional expressions.
III	Operational performance/characteristical evaluation.
III. II	Operational performance evaluation of all system error recording.
III. II. 3. 3. 1	Check for proper termination.
III. II. 3. 3. 2	Check for proper execution.
III. II. 4. 7. 2. 1	Check for proper execution.
III. II. 4. 7. 2. 2	Check for proper execution.
III. II. 4. 7. 2. 3	Check for proper execution.
III. II. 4. 7. 3. 1	Operational analysis.
III. II. 4. 7. 3. 2	Operational analysis.
III. II. 4. 7. 4. 1	Operational analysis.
III. II. 4. 7. 4. 2	Operational analysis.
III. II. 4. 7. 4. 3	Operational analysis.
III. II. 4. 7. 4. 4	Operational analysis.

Reference Number	Measurement
III. II. 4. 7. 5. 1	Analysis of system resources usable.
III. II. 4. 7. 5. 2	Analysis of system resources usable.
III. II. 4. 8. 1	Operational analysis of user-specified limitations.
III. II. 4. 8. 2	Operational analysis of user-specified limitations.
III. II. 4. 8. 3	Operational analysis of user-specified limitations.
III. II. 4. 9. 1	Operational analysis of user-specified limitations.
III. II. 4. 9. 2	Operational analysis of user-specified limitations.
III. II. 4. 9. 3	Operational analysis of user-specified limitations.
III. II. 4. 9. 4. 1	Operational analysis of user-specified limitations.
III. II. 4. 9. 4. 2	Operational analysis of user-specified limitations.
III. III. 1. 1	Operational analysis of error detection and recovery.
III. III. 1. 2	Operational analysis of error detection and recovery.
III. III. 1. 3	Operational analysis of error detection and recovery.
III. III. 1. 4	Operational analysis of error detection and recovery.
III. III. 2	Analysis of error messages.
III. III. 3	Analysis of file backup facilities provided.
III. III. 3. 1	Analysis of file backup facilities provided.
III. III. 3. 2	Analysis of file backup facilities provided.
III. III. 3. 3	Analysis of file backup facilities provided.
III. III. 4. 1	Analysis of processing interrupt facilities.
III. III. 4. 2. 1	Analysis of processing interrupt facilities.
III. III. 4. 2. 2	Analysis of processing interrupt facilities.
III. III. 4. 4	Operational analysis of process termination.
III. III. 4. 5	Operational analysis of process alteration.
III. IV. 9. 1	Operational analysis of automatic destruction capability.
III. IV. 9. 2	Operational analysis of automatic destruction capability.
III. IV. 9. 3	Operational analysis of automatic destruction capability.
III. IV. 11. 1	Analysis of read protection facilities provided.
III. IV. 11. 2	Analysis of read protection facilities provided.
III. IV. 11. 3	Analysis of read protection facilities provided.
III. IV. 11. 4	Analysis of read protection facilities provided.
III. IV. 12. 1	Analysis of write protection facilities provided.
III. IV. 12. 2	Analysis of write protection facilities provided.
III. IV. 12. 3	Analysis of write protection facilities provided.
III. IV. 12. 4	Analysis of write protection facilities provided.
IV	Operational analysis of the host environment interrelationships.
IV. I. 2	Operational analysis of the programming modes provided.
IV. I. 3	Operational analysis of the access methods.
IV. I. 3. 1	Operational analysis of the access methods.
IV. I. 3. 2	Operational analysis of the access methods.
IV. I. 3. 3	Operational analysis of the access methods.
IV. I. 5	Analysis of language form in operational use.
IV. I. 6. 1. 1	Operational analysis of data structure referencing statement.

Reference Number	Measurement
IV. I. 6. 2. 1	Operational analysis of data structure referencing statement.
IV. I. 6. 3. 1	Operational analysis of data structure referencing statement.
IV. I. 6. 4. 1	Operational analysis of data structure referencing statement.
IV. I. 7. 2	Analysis of error handling facilities.
IV. I. 7. 3. 1	Analysis of selection criteria during use.
IV. I. 7. 3. 2	Analysis of selection criteria during use.
IV. I. 7. 3. 2. 1	Analysis of selection criteria during use.
IV. I. 7. 3. 3	Analysis of selection criteria during use.
IV. I. 7. 3. 3. 1	Analysis of selection criteria during use.
IV. I. 7. 3. 3. 2. 1	Analysis of selection criteria during use.
IV. I. 7. 3. 3. 2. 2	Analysis of selection criteria during use.
IV. I. 7. 3. 3. 2. 3	Analysis of selection criteria during use.
IV. I. 7. 3. 3. 3	Analysis of selection criteria during use.
IV. I. 7. 3. 4	Analysis of selection criteria during use.
IV. I. 8. 1. 1	Operational analysis of security features provided.
IV. I. 8. 1. 2	Operational analysis of security features provided.
IV. I. 8. 2	Operational analysis of security features provided.
IV. I. 8. 3. 1	Operational analysis of security features provided.
IV. I. 8. 3. 2	Operational analysis of security features provided.
IV. I. 8. 3. 3	Operational analysis of security features provided.
IV. I. 9	Operational analysis of data manipulation statements.
IV. I. 9. 1	Operational analysis of data manipulation statements.
IV. I. 9. 2	Operational analysis of data manipulation statements.
IV. I. 9. 3	Operational analysis of data manipulation statements.
IV. I. 9. 4	Operational analysis of data manipulation statements.
IV. I. 9. 5	Operational analysis of data manipulation statements.
IV. I. 9. 5. 2	Operational analysis of access statements.
IV. I. 9. 5. 2. 1	Operational analysis of access statements.
IV. I. 9. 5. 3	Operational analysis of these statements.
IV. I. 9. 5. 4	Operational analysis of these statements.
IV. I. 9. 6	Operational analysis of these statements.
IV. I. 9. 6. 1	Operational analysis of these statements.
IV. I. 9. 6. 2	Operational analysis of these statements.
IV. I. 9. 6. 3	Operational analysis of these statements.
IV. I. 9. 6. 4	Operational analysis of these statements.
IV. I. 9. 7	Operational analysis of these statements.
IV. I. 9. 7. 2. 1	Operational analysis of these statements.
IV. I. 9. 7. 2. 2	Operational analysis of these statements.
IV. II. 3. 4. 1	Operational analysis of the sign-off procedure.
IV. II. 3. 4. 2	Operational analysis of the sign-off procedure.
IV. II. 3. 4. 3	Operational analysis of the sign-off procedure.
IV. II. 3. 5. 1. 2. 1	Operational analysis of CRT features.
IV. II. 3. 5. 1. 2. 2	Operational analysis of CRT features.
IV. II. 3. 5. 1. 3. 1	Operational analysis of CRT features.

Reference Number	Measurement
IV. II. 3. 5. 1. 3. 2	Operational analysis of CRT features.
IV. II. 3. 5. 1. 3. 3	Operational analysis of CRT features.
IV. II. 3. 5. 1. 3. 4	Operational analysis of CRT features.
IV. II. 3. 5. 1. 3. 4. 1. 1	Operational analysis of CRT features.
IV. II. 3. 5. 1. 3. 4. 1. 2	Operational analysis of CRT features.
IV. II. 3. 5. 1. 3. 4. 2. 1	Operational analysis of CRT features.
IV. II. 3. 5. 1. 3. 4. 2. 2	Operational analysis of CRT features.
IV. II. 3. 5. 2. 2	Analysis of teletype error correction facilities.
IV. II. 3. 5. 2. 2. 1	Analysis of teletype error correction facilities.
IV. II. 3. 5. 2. 2. 2	Analysis of teletype error correction facilities.
IV. II. 3. 5. 2. 3. 1	Analysis of teletype error correction facilities.
IV. II. 3. 5. 2. 3. 2	Analysis of teletype error correction facilities.
IV. II. 3. 5. 3. 1	Analysis of impact of unusable keyboard keys.
IV. II. 4. 1	Analysis of recovery procedure facilities.
IV. II. 4. 2	Analysis of recovery procedure facilities.
IV. II. 4. 3	Analysis of recovery procedure facilities.
IV. III. 1. 1	Analysis of the uniprogramming environment.
IV. III. 1. 2	Analysis of the multiprogramming environment.
IV. III. 1. 2. 1	Analysis of the multiprogramming environment.
IV. III. 1. 2. 2	Analysis of the multiprogramming environment.
IV. III. 1. 2. 3	Analysis of the multiprogramming environment.
IV. III. 1. 2. 4	Analysis of the multiprogramming environment.
IV. III. 1. 3	Analysis of the multiprocessing environment.
IV. III. 2. 1. 1	Analysis of data base integrity features.
IV. III. 2. 2. 1	Operational evaluation of scheduling/interrupt handling.
IV. III. 2. 2. 2	Operational evaluation of scheduling/interrupt handling.
IV. III. 2. 3. 1. 1	Analysis of features for concurrency of operations.
IV. III. 2. 3. 1. 2	Analysis of features for concurrency of operations.
IV. III. 2. 3. 1. 3	Analysis of features for concurrency of operations.
IV. III. 2. 3. 2. 1	Analysis of features for concurrency of operations.
IV. III. 2. 3. 2. 2	Analysis of features for concurrency of operations.
IV. III. 2. 3. 3. 1	Analysis of features for concurrency of operations.
IV. III. 2. 3. 3. 2	Analysis of features for concurrency of operations.
IV. III. 2. 3. 3. 3	Analysis of features for concurrency of operations.
IV. III. 2. 3. 4. 1	Analysis of features for concurrency of operations.
IV. III. 2. 3. 4. 2	Analysis of features for concurrency of operations.
IV. III. 2. 3. 4. 3	Analysis of features for concurrency of operations.
IV. III. 3	Operational analysis of software facilities.
IV. III. 3. 1. 1	Operational analysis of software facilities.
IV. III. 3. 1. 2	Operational analysis of software facilities.
IV. III. 3. 1. 3	Operational analysis of software facilities.
IV. III. 3. 1. 4	Operational analysis of software facilities.
IV. III. 3. 1. 5	Operational analysis of software facilities.
IV. III. 3. 1. 6	Operational analysis of software facilities.
IV. III. 3. 1. 7	Operational analysis of software facilities.

Reference Number	Measurement
IV. III. 3. 1. 8	Operational analysis of software facilities.
IV. III. 3. 1. 9	Operational analysis of software facilities.
IV. III. 3. 2. 1	Operational analysis of software facilities.
IV. III. 3. 2. 2	Operational analysis of software facilities.
IV. III. 4. 1	Operational analysis of interactive operation.
IV. III. 4. 1. 1. 1	Operational analysis of interactive operation.
IV. III. 4. 1. 1. 2	Operational analysis of interactive operation.
IV. III. 4. 1. 1. 3	Operational analysis of interactive operation.
IV. III. 4. 2	Operational analysis of conversational mode operation.
IV. III. 4. 2. 1	Operational analysis of conversational mode operation.
IV. III. 4. 2. 2	Operational analysis of conversational mode operation.
IV. III. 4. 2. 3	Operational analysis of conversational mode operation.
IV. III. 4. 2. 4	Operational analysis of conversational mode operation.
IV. III. 4. 2. 5	Operational analysis of conversational mode operation.
IV. III. 4. 2. 6	Operational analysis of conversational mode operation.
IV. III. 4. 2. 7	Operational analysis of conversational mode operation.
IV. III. 4. 2. 8	Operational analysis of conversational mode operation.
IV. III. 4. 2. 10. 1	Operational evaluation.
IV. III. 5	Operational analysis of batch process operation.
IV. III. 5. 1. 1	Operational analysis of batch process operation.
IV. III. 5. 1. 2	Operational analysis of batch process operation.
IV. III. 5. 1. 3	Operational analysis of batch process operation.
IV. III. 5. 2	Operational analysis of batch process operation.
IV. III. 5. 2. 1	Operational analysis of batch process operation.
IV. III. 5. 2. 2	Operational analysis of batch process operation.
IV. III. 5. 2. 3	Operational analysis of batch process operation.
IV. III. 5. 2. 4	Operational analysis of batch process operation.
IV. III. 5. 3. 1	Operational analysis of batch process operation.
IV. III. 5. 3. 2	Operational analysis of batch process operation.
IV. III. 5. 3. 3	Operational analysis of batch process operation.
IV. III. 5. 3. 4	Operational analysis of batch process operation.
IV. III. 5. 4	Operational analysis of batch process operation.
IV. III. 5. 5	Operational analysis of batch process operation.
IV. III. 5. 6	Operational analysis of batch process operation.
IV. III. 5. 7	Operational analysis of batch process operation.
IV. III. 5. 8	Operational analysis of batch process operation.
IV. III. 5. 9. 1	Operational analysis of batch process operation.
IV. III. 5. 9. 2	Operational analysis of batch process operation.
IV. III. 5. 10. 1	Operational analysis of batch process operation.
IV. III. 5. 10. 2	Operational analysis of batch process operation.
IV. III. 5. 10. 3	Operational analysis of batch process operation.
IV. III. 5. 11	Operational analysis of batch process operation.
IV. III. 6. 1	Operational analysis of DMS.
IV. III. 6. 2	Operational analysis of DMS.
IV. III. 6. 3	Operational analysis of DMS.
IV. III. 6. 4	Operational analysis of DMS.

MEASUREMENT/TEST PAIRS

Test Technique: Documentation Analysis

Reference Number	Measurement
I. I	Determine types of structure allowable.
I. I. 1. 2. 1. 1	Determine what the system term is.
I. I. 1. 2. 1. 2	Determine the storage representation.
I. I. 1. 2. 1. 3	Determine allowable length.
I. I. 1. 2. 2. 1	Determine what the system term is.
I. I. 1. 2. 2. 2	Determine the storage representation.
I. I. 1. 2. 2. 3	Determine allowable length.
I. I. 1. 3. 1. 1	What is the fixed length.
I. I. 1. 3. 1. 2	Find the allowable length range.
I. I. 1. 3. 2. 2	Find the allowable length range.
I. I. 1. 6. 3	What are the output editing attributes.
I. I. 1. 6. 4	What are the I/O conversion attributes.
I. I. 2. 3. 5. 1	Find whether group identifiers/sequencers are required or optional.
I. I. 2. 3. 5. 2	Find the number of items used as identifiers/sequencers.
I. I. 2. 3. 5. 3	Find whether identifiers/sequencers are ascending/descending sequence.
I. I. 2. 5. 1	Find the number and type of constituent items.
I. I. 2. 5. 2	Find the number and type of constituent items in compound group.
I. I. 2. 5. 4. 3	Find the allowable number of levels of subordination.
I. I. 2. 5. 4. 4	Find the allowable number of dependent groups per parent group.
I. I. 2. 5. 4. 5	Find the allowable number of peer groups at same level of subordination.
I. I. 3. 5. 1	Find allowable number of levels of subordination of composite groups.
I. I. 3. 5. 2	Find allowable number of dependent groups per parent group.
I. I. 3. 5. 3	Find allowable number of peer groups at same level of subordination.
I. I. 3. 5. 4	Find the placement criteria for insertion of a new constituent group occurrence.
I. I. 4. 5. 1	Find the limitation number and type of group/group relationships comprising the entry.
I. I. 4. 5. 2	Find the number of hierarchic levels allowed per entry.
I. I. 4. 5. 3	Find the number of relations in which a dependent group may participate.
I. I. 5. 3. 4. 1	Find the maximum number of synonyms allowable.
I. I. 5. 5. 1	What is the allowable number of files per data base.
I. I. 5. 5. 2	Find the allowable number and type of entries per file.
I. I. 5. 5. 3	Find the limitations on inter-entry relations.

Reference Number	Measurement
I. II. 1. 1	Find what language form is used.
I. II. 3. 6. 1	Find which attributes may be deleted for an item.
I. II. 3. 6. 2	Find which attributes may be deleted for a group.
I. II. 3. 6. 3	Find which attributes may be deleted for a group relation.
I. II. 3. 6. 4	Find which attributes may be deleted for an entry.
I. II. 3. 6. 5	Find which attributes may be deleted for a file.
I. II. 3. 7	Find which attributes may be expanded or modified.
I. II. 3. 8	Find which attributes may be deleted or replaced.
I. II. 4. 7. 3	Find the sequence of constituent definitions.
I. III	What is the allowable storage structure.
I. III. 1. 2. 1. 1	What sequential devices are available.
I. III. 1. 2. 1. 2	What direct access devices are available.
I. III. 1. 2. 4	What control does the user have over index arrangement.
I. III. 4	What storage access methods are available.
II	What data manipulation functions are available.
II. I. 2. 1. 5	What diagnostics are provided.
II. I. 3. 1. 2. 3	What other physical media is acceptable.
II. I. 3. 2. 5	What other system processors are available.
II. I. 3. 6	What are the multi-file input capabilities.
II. I. 4. 5	What monitoring and error detection facilities are provided.
II. II. 3. 2. 1	What format(s) is used.
II. II. 3. 2. 6	What data validation features are provided.
II. II. 3. 2. 7	What data editing and transformation features are provided.
II. II. 4. 2. 1	Find the format used.
II. II. 4. 4	Find the data validation features available.
II. II. 4. 5	Find the editing and transformation features available.
II. II. 6. 1	Find what system detected errors are provided.
II. II. 6. 2	Find what operating system errors are provided.
II. II. 6. 8	Find what system statistics are provided.
II. II. 6. 9	Find what audit trail features are provided.
II. II. 6. 10	Find what backup file features are available.
II. III. 1. 1. 1	Find what operators are available.
II. III. 1. 1. 2	Find what logical connectors are available.
II. III. 1. 2. 1	Find how complex expressions can be.
II. III. 1. 2. 2	Find how many levels of nesting are provided.
II. III. 2. 2	Find what data selection statements are available.
II. III. 3. 2	Find what data extraction statements are available.
II. III. 7. 3. 4	Find the number of lines per page provided.
II. III. 8. 3. 8	Find what standard options are provided.
II. IV. 5. 3. 3	Determine how many conditional expressions can be combined directly.

Reference Number	Measurement
II. IV. 5. 3. 4	Determine maximum number of levels of nesting using parentheses.
II. IV. 5. 3. 7	Find what the precedence rules for logical connectors within parentheses are.
III. I. 1. 1	Find how many recording categories are provided.
III. II	Find out what types of error recording are provided.
III. IV. 5	Determine the maximum number of access categories within each security level.
IV. I. 7. 2	Determine what type of error handling is provided.
IV. I. 7. 3. 2. 1	Find what conditional expression capabilities are available for selection.
IV. I. 7. 3. 3. 1	Find what logical and relational operators are provided.
IV. II. 1. 1	Find what the processor requirements are.
IV. II. 1. 2. 1	Find what the minimum memory requirements are.
IV. II. 1. 2. 2	Find what the minimum memory requirements are.
IV. II. 1. 3	Find what the required hardware options are.
IV. II. 2. 1. 1. 1	Determine what the tape drive requirements are.
IV. II. 2. 1. 1. 2	Determine the direct access device requirements.
IV. II. 2. 1. 2. 1	Determine what the tape drive requirements are.
IV. II. 2. 1. 2. 2	Determine the direct access device requirements.
IV. II. 3. 1. 1	Find the maximum number of on-line consoles or terminals to be connected.
IV. II. 3. 1. 2	Find the maximum number of active consoles.
IV. II. 3. 1. 3	Find the maximum number of on-line users.
IV. II. 3. 2	Determine what the machine interface is.
IV. II. 3. 3. 1	Find what the manual start-up procedure is.
IV. II. 3. 3. 2	Find what the automatic start-up procedure is.
IV. II. 3. 5. 1. 3. 4. 3. 1	Determine the number of lines per display.
IV. II. 3. 5. 1. 3. 4. 3. 2	Determine the number of characters per line or display.
IV. II. 3. 5. 1. 3. 4. 4	Determine what the available character set is.
IV. II. 3. 5. 2. 1	Determine the speed in characters per second on the teletype.
IV. II. 3. 5. 3. 1	Find what the system reserved keyboard characters are.
IV. II. 3. 5. 3. 2	Find the number of special command keys and their definition.
IV. III. 3. 1. 6	Determine what the additional software feature facilities are.

Reference Number	Measurement
IV. III. 4. 1. 1. 3	Find which DMS functions are available in the interactive mode.
IV. III. 4. 2. 1	Find which DMS functions are available in the conversational mode.
IV. III. 5. 11	Determine the maximum number of users that can simultaneously operate remotely.
IV. IV. 1. 1	Determine if the documentation is adequate.
IV. IV. 1. 2	Determine if the documentation is adequate.
IV. IV. 1. 3	Determine if the documentation is adequate.
IV. IV. 1. 4	Determine if the documentation is adequate.
IV. IV. 2. 1. 1	Determine if the documentation is adequate.
IV. IV. 2. 1. 2	Determine if the documentation is adequate.
IV. IV. 2. 1. 3	Determine if the documentation is adequate.
IV. IV. 2. 2. 1	Determine if the documentation is adequate.
IV. IV. 2. 2. 2	Determine if the documentation is adequate.
IV. IV. 2. 2. 3	Determine if the documentation is adequate.

MEASUREMENT/TEST PAIRS

Test Technique: Numerical Scoring Methods

Reference Number	Measurement
I. I. 1. 1	Yes/no.
I. I. 1. 2. 1. 1	Yes/no.
I. I. 1. 2. 1. 2	Rating scheme.
I. I. 1. 2. 1. 3	Rating scheme.
I. I. 1. 2. 1. 4	Yes/no.
I. I. 1. 2. 2. 1	Yes/no.
I. I. 1. 2. 2. 2	Rating scheme.
I. I. 1. 2. 2. 3	Rating scheme.
I. I. 1. 2. 2. 4	Yes/no.
I. I. 1. 2. 3. 1	Yes/no.
I. I. 1. 2. 3. 2	Yes/no.
I. I. 1. 2. 3. 3	Yes/no.
I. I. 1. 3. 1	Yes/no.
I. I. 1. 3. 1. 1	Yes/no.
I. I. 1. 3. 1. 2	Rating scheme.
I. I. 1. 3. 1. 3. 1	Yes/no.
I. I. 1. 3. 1. 3. 2	Yes/no, rating scheme.
I. I. 1. 3. 1. 3. 3	Yes/no.
I. I. 1. 3. 1. 4	Rating scheme, yes/no.
I. I. 1. 3. 1. 5	Yes/no.
I. I. 1. 3. 1. 6	Yes/no.
I. I. 1. 3. 2. 1	Yes/no.
I. I. 1. 3. 2. 2	Yes/no, rating scheme.
I. I. 1. 3. 2. 3. 1	Yes/no.
I. I. 1. 3. 2. 3. 2	Yes/no.
I. I. 1. 3. 2. 3. 4	Yes/no, rating scheme.
I. I. 1. 3. 2. 5	Yes/no.
I. I. 1. 3. 2. 6	Yes/no.
I. I. 1. 4. 1	Yes/no.
I. I. 1. 4. 2	Yes/no.
I. I. 1. 4. 3	Yes/no.
I. I. 1. 4. 4	Yes/no.
I. I. 1. 4. 5. 1	Yes/no.
I. I. 1. 4. 5. 2	Yes/no.
I. I. 1. 4. 6	Yes/no.
I. I. 1. 5. 1. 1	Yes/no.
I. I. 1. 5. 1. 2	Yes/no.
I. I. 1. 5. 2. 1	Yes/no.
I. I. 1. 5. 2. 2	Yes/no.
I. I. 1. 6. 1	Yes/no, rating scheme.
I. I. 1. 6. 2	Yes/no, rating scheme.

Reference Number	Measurement
I. I. 1. 6. 3	Yes/no, rating scheme.
I. I. 1. 6. 4	Yes/no, rating scheme.
I. I. 1. 6. 5	Yes/no.
I. I. 1. 7. 1	Yes/no.
I. I. 1. 7. 2	Yes/no.
I. I. 1. 7. 3	Yes/no.
I. I. 1. 7. 4	Yes/no.
I. I. 1. 7. 5	Yes/no.
I. I. 2. 1	Yes/no.
I. I. 2. 2. 1	Yes/no.
I. I. 2. 2. 2	Yes/no.
I. I. 2. 2. 3	Yes/no.
I. I. 2. 2. 4	Yes/no.
I. I. 2. 3. 1	Yes/no, rating scheme.
I. I. 2. 3. 2	Rating scheme.
I. I. 2. 3. 3	Yes/no.
I. I. 2. 3. 4	Yes/no.
I. I. 2. 3. 5. 1	Yes/no.
I. I. 2. 3. 5. 2	Rating scheme.
I. I. 2. 3. 5. 3	Yes/no, rating scheme.
I. I. 2. 4. 1. 1	Yes/no.
I. I. 2. 4. 1. 2	Yes/no.
I. I. 2. 4. 2. 1	Yes/no.
I. I. 2. 4. 2. 2	Yes/no.
I. I. 2. 5. 1	Rating scheme.
I. I. 2. 5. 2	Rating scheme.
I. I. 2. 5. 3	Rating scheme.
I. I. 2. 5. 4	Rating scheme.
I. I. 2. 5. 4. 1	Yes/no.
I. I. 2. 5. 4. 2	Yes/no.
I. I. 2. 5. 4. 3	Rating scheme.
I. I. 2. 5. 4. 4	Rating scheme.
I. I. 2. 5. 4. 5	Rating scheme.
I. I. 2. 6. 1	Yes/no.
I. I. 2. 6. 2	Yes/no.
I. I. 3. 1	Yes/no.
I. I. 3. 2. 1	Yes/no.
I. I. 3. 2. 2. 1	Yes/no.
I. I. 3. 2. 2. 2	Yes/no.
I. I. 3. 3. 1	Yes/no.
I. I. 3. 3. 2	Yes/no.
I. I. 3. 3. 3	Yes/no.
I. I. 3. 3. 4	Yes/no.
I. I. 3. 4. 1. 1	Yes/no.
I. I. 3. 4. 1. 2	Yes/no.

Reference Number	Measurement
I. I. 3. 4. 2. 1	Yes/no.
I. I. 3. 4. 2. 2	Yes/no.
I. I. 3. 5. 1	Rating scheme.
I. I. 3. 5. 2	Rating scheme.
I. I. 3. 5. 3	Rating scheme.
I. I. 3. 5. 4	Rating scheme.
I. I. 3. 6. 1	Rating scheme.
I. I. 4. 1	Yes/no.
I. I. 4. 2. 1	Yes/no.
I. I. 4. 2. 2	Yes/no.
I. I. 4. 3. 1	Yes/no.
I. I. 4. 3. 2	Yes/no.
I. I. 4. 3. 3	Yes/no.
I. I. 4. 3. 4	Yes/no.
I. I. 4. 3. 5	Yes/no.
I. I. 4. 4. 1. 1	Yes/no.
I. I. 4. 4. 1. 2	Yes/no.
I. I. 4. 4. 2. 1	Yes/no.
I. I. 4. 4. 2. 2	Yes/no.
I. I. 4. 5. 1	Yes/no.
I. I. 4. 5. 2	Yes/no.
I. I. 4. 5. 3	Yes/no.
I. I. 4. 6. 1	Yes/no.
I. I. 4. 6. 2	Yes/no.
I. I. 5. 1	Yes/no.
I. I. 5. 2. 1	Yes/no.
I. I. 5. 2. 2. 1	Yes/no.
I. I. 5. 2. 2. 2	Yes/no.
I. I. 5. 3. 1	Yes/no.
I. I. 5. 3. 2	Yes/no.
I. I. 5. 3. 3	Yes/no.
I. I. 5. 3. 4	Yes/no.
I. I. 5. 3. 4. 1	Rating scheme.
I. I. 5. 3. 4. 2	Yes/no.
I. I. 5. 4. 1. 1	Yes/no.
I. I. 5. 4. 1. 2	Yes/no.
I. I. 5. 5. 1	Rating scheme.
I. I. 5. 5. 2	Rating scheme.
I. I. 5. 5. 3	Yes/no, rating scheme.
I. I. 5. 6. 1	Yes/no.
I. I. 5. 6. 2	Yes/no.
I. I. 5. 6. 3	Yes/no.
I. I. 5. 6. 4	Yes/no.
I. II. 1. 1	Rating scheme.
I. II. 1. 2	Yes/no.

Reference Number	Measurement
I. II. 1. 3	Yes/no.
I. II. 2. 1	Yes/no.
I. II. 2. 2	Yes/no.
I. II. 2. 3	Yes/no.
I. II. 2. 4	Yes/no.
I. II. 3. 1	Yes/no.
I. II. 3. 2	Yes/no.
I. II. 3. 3	Yes/no.
I. II. 3. 4	Yes/no.
I. II. 3. 5	Yes/no.
I. II. 3. 6. 1	Yes/no.
I. II. 3. 6. 2	Yes/no.
I. II. 3. 6. 3	Yes/no.
I. II. 3. 6. 4	Yes/no.
I. II. 3. 6. 5	Yes/no.
I. II. 3. 7	Yes/no.
I. II. 3. 8	Yes/no.
I. II. 4. 1. 1. 1	Yes/no.
I. II. 4. 1. 1. 2	Yes/no.
I. II. 4. 1. 1. 3	Yes/no.
I. II. 4. 1. 2. 1. 1	Yes/no.
I. II. 4. 1. 2. 1. 2	Yes/no.
I. II. 4. 2. 1	Yes/no.
I. II. 4. 2. 2. 1	Yes/no, rating scheme.
I. II. 4. 2. 2. 2	Yes/no.
I. II. 4. 2. 3. 1	Yes/no.
I. II. 4. 2. 3. 2. 1	Yes/no.
I. II. 4. 2. 3. 2. 2	Yes/no.
I. II. 4. 2. 3. 3	Yes/no.
I. II. 4. 2. 3. 4	Yes/no.
I. II. 4. 2. 3. 4. 1	Yes/no.
I. II. 4. 2. 3. 4. 2	Yes/no.
I. II. 4. 2. 3. 4. 3. 1	Yes/no.
I. II. 4. 2. 3. 4. 3. 2	Yes/no.
I. II. 4. 2. 3. 4. 3. 3	Yes/no.
I. II. 4. 2. 3. 4. 3. 4	Yes/no.
I. II. 4. 2. 3. 4. 3. 5	Yes/no.
I. II. 4. 2. 3. 4. 3. 6	Yes/no.
I. II. 4. 2. 3. 4. 3. 7	Yes/no.
I. II. 4. 2. 3. 4. 3. 8	Yes/no.
I. II. 4. 2. 3. 4. 3. 9	Yes/no.
I. II. 4. 2. 3. 4. 3. 10	Yes/no.
I. II. 4. 2. 3. 4. 3. 11	Yes/no.

Reference Number	Measurement
I. II. 4. 2. 3. 4. 3. 12	Yes/no.
I. II. 4. 2. 3. 4. 3. 13	Yes/no.
I. II. 4. 2. 3. 4. 3. 14	Yes/no.
I. II. 4. 2. 3. 4. 3. 15	Yes/no.
I. II. 4. 2. 3. 4. 3. 16	Yes/no.
I. II. 4. 2. 3. 4. 3. 17	Yes/no.
I. II. 4. 2. 3. 4. 3. 18	Yes/no.
I. II. 4. 2. 3. 4. 3. 19	Yes/no.
I. II. 4. 2. 3. 4. 3. 20	Yes/no.
I. II. 4. 2. 4	Yes/no.
I. II. 4. 2. 5	Yes/no.
I. II. 4. 3. 1	Yes/no.
I. II. 4. 3. 2	Yes/no.
I. II. 4. 3. 3	Yes/no.
I. II. 4. 4. 1. 1	Yes/no.
I. II. 4. 4. 1. 2	Yes/no.
I. II. 4. 4. 2. 1	Yes/no.
I. II. 4. 4. 2. 2	Yes/no.
I. II. 4. 4. 2. 3	Yes/no.
I. II. 4. 4. 3	Yes/no.
I. II. 4. 4. 4. 1. 1	Yes/no.
I. II. 4. 4. 4. 1. 2	Yes/no.
I. II. 4. 4. 5. 1	Rating scheme.
I. II. 4. 4. 5. 2	Yes/no.
I. II. 4. 4. 6	Yes/no.
I. II. 4. 4. 6. 1	Yes/no, rating scheme.
I. II. 4. 4. 7	Yes/no.
I. II. 4. 4. 8	Yes/no.
I. II. 4. 4. 8. 1	Yes/no.
I. II. 4. 4. 8. 2	Yes/no.
I. II. 4. 5. 1	Yes/no.
I. II. 4. 5. 2. 1. 1	Yes/no.
I. II. 4. 5. 2. 1. 2	Yes/no.
I. II. 4. 5. 3	Yes/no.
I. II. 4. 5. 4	Yes/no.
I. II. 4. 5. 5	Yes/no.
I. II. 4. 5. 6	Yes/no.

Reference Number	Measurement
I. II. 4. 6. 1	Yes/no.
I. II. 4. 6. 2. 1. 1	Yes/no.
I. II. 4. 6. 2. 1. 2	Yes/no.
I. II. 4. 6. 3. 1	Yes/no.
I. II. 4. 6. 3. 2	Yes/no.
I. II. 4. 6. 4	Yes/no.
I. II. 4. 7. 1	Yes/no.
I. II. 4. 7. 2. 1. 1	Yes/no.
I. II. 4. 7. 2. 1. 2	Yes/no.
I. II. 4. 7. 3	Rating scheme.
I. III. 1. 1. 1	Yes/no.
I. III. 1. 1. 2	Yes/no.
I. III. 1. 1. 3	Yes/no.
I. III. 1. 1. 4	Yes/no.
I. III. 1. 2. 1. 1	Yes/no, rating scheme.
I. III. 1. 2. 1. 2	Yes/no, rating scheme.
I. III. 1. 2. 2	Yes/no.
I. III. 1. 2. 2. 1	Yes/no.
I. III. 1. 2. 2. 2	Yes/no.
I. III. 1. 2. 3. 1	Yes/no.
I. III. 1. 2. 3. 2	Yes/no.
I. III. 1. 2. 3. 3	Yes/no.
I. III. 1. 2. 3. 4	Yes/no.
I. III. 1. 2. 3. 5	Yes/no.
I. III. 1. 2. 3. 6	Yes/no.
I. III. 1. 2. 3. 7	Yes/no.
I. III. 1. 2. 4	Yes/no, possibly rating scheme.
I. III. 1. 2. 5	Yes/no.
I. III. 2. 1	Yes/no.
I. III. 2. 2	Yes/no.
I. III. 2. 2. 1	Yes/no.
I. III. 3. 1	Yes/no.
I. III. 3. 2. 1	Yes/no.
I. III. 3. 2. 2	Yes/no.
I. III. 3. 2. 2. 1	Yes/no.
I. III. 3. 2. 2. 2	Yes/no.
I. III. 3. 2. 2. 3	Yes/no.
I. III. 3. 2. 2. 4. 1	Yes/no.
I. III. 3. 2. 2. 4. 2	Yes/no.
I. III. 3. 2. 2. 4. 3	Yes/no.
I. III. 3. 2. 2. 5	Yes/no.
I. III. 3. 2. 2. 6	Yes/no.
I. III. 4. 1	Yes/no.
I. III. 4. 1. 1	Yes/no.
I. III. 4. 1. 1. 1	Yes/no.

Reference Number	Measurement
I. III. 4. 1. 2. 1	Yes/no.
I. III. 4. 1. 2. 2	Yes/no.
I. III. 4. 1. 2. 3	Yes/no.
I. III. 4. 2	Yes/no.
I. III. 4. 2. 1	Yes/no.
I. III. 4. 2. 2	Yes/no.
I. III. 4. 2. 2. 1	Yes/no.
I. III. 4. 2. 3	Yes/no.
I. III. 4. 2. 4	Yes/no.
II. I. 1. 1	Yes/no.
II. I. 1. 2	Yes/no.
II. I. 1. 3	Yes/no.
II. I. 1. 4	Yes/no.
II. I. 1. 5	Yes/no.
II. I. 1. 6	Yes/no.
II. I. 2. 1. 1	Yes/no.
II. I. 2. 1. 2	Yes/no.
II. I. 2. 1. 3	Yes/no.
II. I. 2. 1. 4	Yes/no.
II. I. 2. 1. 5	Yes/no.
II. I. 2. 2	Yes/no.
II. I. 2. 3. 1	Yes/no.
II. I. 2. 3. 2	Yes/no.
II. I. 2. 3. 3. 1	Yes/no.
II. I. 2. 3. 3. 2	Yes/no.
II. I. 2. 3. 3. 3	Yes/no.
II. I. 3	Yes/no.
II. I. 3. 1. 1	Yes/no.
II. I. 3. 1. 2. 1	Yes/no.
II. I. 3. 1. 2. 2	Yes/no.
II. I. 3. 1. 2. 3	Yes/no, rating scheme.
II. I. 3. 2. 1	Yes/no.
II. I. 3. 2. 2	Yes/no.
II. I. 3. 2. 3	Yes/no.
II. I. 3. 2. 4	Yes/no.
II. I. 3. 2. 5	Yes/no, rating scheme.
II. I. 3. 3	Yes/no, rating scheme.
II. I. 3. 4	Yes/no.
II. I. 3. 5. 1. 1	Yes/no.
II. I. 3. 5. 1. 2	Yes/no.
II. I. 3. 5. 1. 3	Yes/no.
II. I. 3. 5. 2	Yes/no.
II. I. 3. 5. 3. 1	Yes/no.
II. I. 3. 5. 3. 2	Yes/no.
II. I. 3. 5. 3. 3	Yes/no.

Reference Number	Measurement
II. I. 3. 5. 4. 1	Yes/no.
II. I. 3. 5. 4. 2	Yes/no.
II. I. 3. 5. 4. 3	Yes/no.
II. I. 3. 5. 5. 1	Yes/no.
II. I. 3. 5. 5. 2	Yes/no.
II. I. 3. 5. 5. 3	Yes/no.
II. I. 3. 5. 5. 4	Yes/no.
II. I. 3. 5. 5. 5	Yes/no.
II. I. 3. 5. 5. 6	Yes/no.
II. I. 3. 5. 5. 7	Yes/no.
II. I. 3. 5. 6. 1	Yes/no.
II. I. 3. 5. 6. 2	Yes/no.
II. I. 3. 5. 6. 3	Yes/no.
II. I. 3. 5. 6. 4	Yes/no.
II. I. 3. 5. 6. 5	Yes/no.
II. I. 3. 5. 6. 6	Yes/no.
II. I. 3. 5. 6. 7	Yes/no.
II. I. 3. 5. 7. 1	Yes/no.
II. I. 3. 5. 7. 2	Yes/no.
II. I. 3. 5. 7. 3	Yes/no.
II. I. 3. 5. 7. 4	Yes/no.
II. I. 3. 6	Yes/no, rating scheme.
II. I. 3. 7. 1	Yes/no.
II. I. 3. 7. 2	Yes/no.
II. I. 3. 7. 3	Yes/no.
II. I. 3. 7. 4	Yes/no.
II. I. 4. 1	Yes/no.
II. I. 4. 2	Yes/no.
II. I. 4. 3. 1	Yes/no.
II. I. 4. 3. 2	Yes/no.
II. I. 4. 3. 3	Yes/no.
II. I. 4. 4. 1	Yes/no.
II. I. 4. 4. 2	Yes/no.
II. I. 4. 4. 3	Yes/no.
II. I. 4. 5	Yes/no, possibly a r[illegible]g scheme.
II. I. 5. 1	Yes/no.
II. I. 5. 2	Yes/no.
II. I. 5. 3	Yes/no.
II. I. 5. 4	Yes/no.
II. I. 5. 5	Yes/no.
II. I. 5. 6	Yes/no.
II. I. 5. 7	Yes/no.
II. I. 5. 8	Yes/no.
II. II. 1. 1	Yes/no.
II. II. 1. 2	Yes/no.

Reference Number	Measurement
II. II. 1. 3	Yes/no.
II. II. 1. 4	Yes/no.
II. II. 2. 1	Rating scheme.
II. II. 2. 2. 1. 1	Yes/no.
II. II. 2. 2. 1. 2	Yes/no.
II. II. 2. 2. 1. 3	Yes/no.
II. II. 2. 2. 1. 4	Yes/no.
II. II. 2. 2. 1. 4. 1	Yes/no.
II. II. 2. 2. 1. 4. 2	Yes/no.
II. II. 2. 2. 1. 5	Yes/no.
II. II. 2. 2. 1. 5. 1	Yes/no.
II. II. 2. 2. 1. 5. 2	Yes/no.
II. II. 2. 2. 1. 6	Yes/no.
II. II. 2. 2. 2. 1	Yes/no.
II. II. 2. 2. 2. 2	Yes/no.
II. II. 2. 2. 3. 1	Yes/no.
II. II. 2. 2. 3. 2	Yes/no.
II. II. 2. 3. 1	Yes/no, rating scheme.
II. II. 2. 4. 1	Yes/no.
II. II. 2. 4. 1. 1	Yes/no.
II. II. 2. 4. 2	Yes/no.
II. II. 2. 4. 3	Yes/no.
II. II. 3. 1	Yes/no, rating scheme.
II. II. 3. 2. 1	Yes/no, rating scheme.
II. II. 3. 2. 2	Yes/no.
II. II. 3. 2. 3	Yes/no.
II. II. 3. 2. 4. 1	Yes/no.
II. II. 3. 2. 4. 2	Yes/no.
II. II. 3. 2. 5	Yes/no
II. II. 3. 2. 6	Yes/no, possibly rating scheme.
II. II. 3. 2. 7	Yes/no, possibly rating scheme.
II. II. 4. 1	Yes/no, rating scheme.
II. II. 4. 2. 1	Yes/no.
II. II. 4. 2. 2	Yes/no.
II. II. 4. 2. 3	Yes/no.
II. II. 4. 2. 4	Yes/no.
II. II. 4. 3	Yes/no.
II. II. 4. 4	Yes/no, possibly rating scheme.
II. II. 4. 5	Yes/no, rating scheme.
II. II. 5. 1. 1. 1	Yes/no.
II. II. 5. 1. 1. 2	Yes/no.
II. II. 5. 1. 1. 3	Yes/no.
II. II. 5. 1. 1. 4	Yes/no.
II. II. 5. 1. 2. 1. 1	Yes/no.
II. II. 5. 1. 2. 2. 1	Yes/no.
II. II. 5. 2. 1	Yes/no.

Reference Number	Measurement
II. II. 5. 2. 1. 1	Yes/no, rating scheme.
II. II. 5. 2. 1. 1. 1	Yes/no, rating scheme.
II. II. 5. 2. 1. 1. 2	Yes/no.
II. II. 5. 3. 1	Yes/no, rating scheme.
II. II. 5. 3. 2. 1	Yes/no.
II. II. 5. 3. 2. 2	Yes/no.
II. II. 5. 3. 2. 3	Rating scheme.
II. II. 5. 3. 2. 4	Yes/no.
II. II. 5. 3. 2. 5	Yes/no.
II. II. 5. 3. 2. 6	Yes/no.
II. II. 5. 3. 3. 1	Yes/no.
II. II. 5. 3. 3. 1. 1. 1	Yes/no.
II. II. 5. 3. 3. 1. 1. 2	Yes/no.
II. II. 5. 3. 3. 1. 1. 3	Yes/no.
II. II. 5. 3. 3. 1. 1. 4	Yes/no.
II. II. 5. 3. 3. 1. 1. 5	Yes/no.
II. II. 5. 3. 3. 1. 1. 6	Yes/no.
II. II. 5. 3. 3. 1. 2. 1	Yes/no.
II. II. 5. 3. 3. 1. 2. 2	Yes/no.
II. II. 5. 3. 3. 1. 2. 3	Yes/no.
II. II. 5. 3. 3. 1. 2. 4	Yes/no.
II. II. 5. 3. 3. 1. 2. 5	Yes/no.
II. II. 5. 3. 3. 1. 2. 6	Yes/no.
II. II. 5. 3. 3. 1. 3. 1	Yes/no.
II. II. 5. 3. 3. 1. 3. 2	Yes/no.
II. II. 5. 3. 3. 1. 3. 3	Yes/no.
II. II. 5. 3. 3. 1. 3. 4	Yes/no.
II. II. 5. 3. 3. 1. 3. 5	Yes/no.
II. II. 5. 3. 3. 1. 3. 6	Yes/no.
II. II. 5. 3. 3. 2. 1	Yes/no, rating scheme.
II. II. 5. 3. 3. 3. 1	Yes/no, rating scheme.
II. II. 5. 3. 3. 4. 1	Yes/no, rating scheme.
II. II. 6. 1	Yes/no.
II. II. 6. 2	Yes/no.
II. II. 6. 3	Yes/no.
II. II. 6. 4	Yes/no.
II. II. 6. 5	Yes/no.
II. II. 6. 6	Yes/no.
II. II. 6. 7	Yes/no.
II. II. 6. 8	Yes/no.
II. II. 6. 9	Yes/no.
II. II. 6. 10	Yes/no.
II. II. 6. 11. 1	Yes/no.
II. II. 6. 11. 2	Yes/no.

Reference Number	Measurement
II. III. 1. 1	Yes/no, rating scheme.
II. III. 1. 1. 1	Yes/no.
II. III. 1. 1. 2	Yes/no.
II. III. 1. 2. 1	Yes/no.
II. III. 1. 2. 2	Yes/no, possibly rating scheme.
II. III. 1. 2. 3	Yes/no.
II. III. 1. 2. 4	Yes/no.
II. III. 1. 3	Yes/no.
II. III. 1. 4. 1	Yes/no.
II. III. 1. 4. 2	Yes/no.
II. III. 1. 4. 3	Yes/no.
II. III. 1. 4. 4	Yes/no.
II. III. 1. 4. 5	Yes/no.
II. III. 1. 4. 6	Yes/no.
II. III. 1. 4. 7	Yes/no.
II. III. 2. 1	Yes/no,
II. III. 2. 2	Yes/no, possibly rating scheme.
II. III. 2. 3. 1	Yes/no.
II. III. 2. 3. 1. 1	Yes/no.
II. III. 2. 3. 1. 2	Yes/no.
II. III. 2. 3. 2	Yes/no.
II. III. 2. 4	Yes/no.
II. III. 2. 5	Yes/no.
II. III. 2. 6	Yes/no.
II. III. 2. 6. 1	Yes/no.
II. III. 2. 6. 2	Yes/no.
II. III. 2. 6. 3	Yes/no.
II. III. 2. 6. 4	Yes/no.
II. III. 2. 6. 5	Yes/no.
II. III. 2. 6. 6	Yes/no.
II. III. 2. 6. 7	Yes/no.
II. III. 3	Yes/no.
II. III. 3. 1	Yes/no.
II. III. 3. 2	Yes/no, possibly a rating scheme.
II. III. 3. 3	Yes/no.
II. III. 3. 4. 1	Yes/no.
II. III. 3. 4. 2	Yes/no.
II. III. 3. 4. 3	Yes/no.
II. III. 3. 4. 4	Yes/no.
II. III. 3. 4. 5	Yes/no.
II. III. 3. 4. 6	Yes/no.
II. III. 3. 5	Yes/no.
II. III. 3. 6. 1	Yes/no.
II. III. 3. 6. 2	Yes/no.
II. III. 3. 7	Yes/no.
II. III. 3. 8	Yes/no, possibly a rating scheme of other devices.
II. III. 3. 9	Yes/no.

Reference Number		Measurement
II. III. 4. 1	Yes/no.	
II. III. 4. 2	Yes/no.	
II. III. 4. 3	Yes/no.	
II. III. 4. 4. 1	Yes/no.	
II. III. 4. 4. 2	Yes/no.	
II. III. 4. 5. 1	Yes/no.	
II. III. 4. 5. 2	Yes/no.	
II. III. 4. 6. 1	Yes/no.	
II. III. 4. 6. 2	Yes/no.	
II. III. 4. 6. 3	Yes/no.	
II. III. 5. 1	Yes/no.	
II. III. 5. 1. 1	Yes/no.	
II. III. 5. 2	Yes/no.	
II. III. 5. 2. 1	Yes/no.	
II. III. 6. 1. 1	Yes/no.	
II. III. 6. 1. 2	Yes/no.	
II. III. 6. 2. 1	Yes/no.	
II. III. 6. 2. 2	Yes/no.	
II. III. 6. 3. 1	Yes/no.	
II. III. 6. 3. 2	Yes/no.	
II. III. 6. 3. 3	Yes/no.	
II. III. 6. 3. 4	Yes/no.	
II. III. 6. 4. 1	Yes/no.	
II. III. 6. 4. 2	Yes/no.	
II. III. 6. 5. 1	Yes/no.	
II. III. 6. 5. 2	Yes/no.	
II. III. 6. 5. 3	Yes/no.	
II. III. 6. 5. 4	Yes/no.	
II. III. 6. 6. 1. 1	Yes/no.	
II. III. 6. 6. 1. 2	Yes/no.	
II. III. 6. 6. 1. 3	Yes/no.	
II. III. 6. 6. 2. 1	Yes/no.	
II. III. 6. 6. 2. 2	Yes/no.	
II. III. 6. 6. 2. 3	Yes/no.	
II. III. 6. 6. 2. 4	Yes/no.	
II. III. 6. 6. 3. 1	Yes/no.	
II. III. 6. 6. 3. 2	Yes/no.	
II. III. 6. 6. 4	Yes/no.	
II. III. 6. 6. 4. 1	Yes/no.	
II. III. 7. 1. 1	Yes/no.	
II. III. 7. 1. 2	Yes/no.	
II. III. 7. 2. 1	Yes/no.	
II. III. 7. 2. 2	Yes/no.	
II. III. 7. 2. 3. 1	Yes/no.	
II. III. 7. 2. 3. 2	Yes/no.	
II. III. 7. 2. 4	Yes/no.	

Reference Number	Measurement
II. III. 7. 3. 1	Yes/no.
II. III. 7. 3. 2	Yes/no.
II. III. 7. 3. 3	Yes/no.
II. III. 7. 3. 4	Yes/no.
II. III. 7. 3. 5	Yes/no, rating scheme.
II. III. 7. 3. 6	Yes/no, rating scheme.
II. III. 7. 4. 1	Yes/no.
II. III. 7. 4. 2	Yes/no.
II. III. 7. 4. 2. 1	Yes/no.
II. III. 7. 4. 2. 2	Yes/no.
II. III. 7. 4. 3	Yes/no.
II. III. 7. 4. 4	Yes/no.
II. III. 7. 4. 5. 1	Yes/no.
II. III. 7. 4. 5. 2	Yes/no.
II. III. 7. 4. 5. 3	Yes/no.
II. III. 7. 4. 6	Yes/no, possibly a rating scheme.
II. III. 7. 4. 7	Yes/no.
II. III. 7. 5. 1	Yes/no.
II. III. 7. 5. 2	Yes/no.
II. III. 7. 5. 3	Yes/no.
II. III. 7. 5. 4	Yes/no.
II. III. 7. 5. 5	Yes/no.
II. III. 7. 5. 6	Yes/no.
II. III. 7. 5. 7	Yes/no.
II. III. 7. 5. 8	Yes/no.
II. III. 7. 5. 9	Yes/no.
II. III. 7. 5. 9. 1	Yes/no.
II. III. 7. 5. 9. 2	Yes/no.
II. III. 7. 5. 9. 3	Yes/no.
II. III. 7. 5. 9. 4	Yes/no.
II. III. 7. 5. 10	Yes/no, possibly a rating scheme.
II. III. 7. 5. 11	Yes/no, possibly a rating scheme.
II. III. 7. 5. 12	Yes/no.
II. III. 7. 5. 13	Yes/no.
II. III. 7. 5. 14	Yes/no.
II. III. 8. 1	Yes/no.
II. III. 8. 1. 1	Yes/no.
II. III. 8. 1. 2	Yes/no.
II. III. 8. 2	Yes/no.
II. III. 8. 2. 1	Yes/no.
II. III. 8. 2. 2	Yes/no.
II. III. 8. 2. 3	Yes/no.
II. III. 8. 2. 4	Yes/no.
II. III. 8. 2. 5	Yes/no.
II. III. 8. 3. 1	Yes/no.
II. III. 8. 3. 2	Yes/no.

Reference Number	Measurement
II. III. 8. 3. 3	Yes/no.
II. III. 8. 3. 4	Yes/no.
II. III. 8. 3. 5	Yes/no.
II. III. 8. 3. 6. 1	Yes/no.
II. III. 8. 3. 6. 2	Yes/no.
II. III. 8. 3. 6. 3	Yes/no.
II. III. 8. 3. 6. 4	Yes/no.
II. III. 8. 3. 6. 5	Yes/no.
II. III. 8. 3. 7. 1	Yes/no.
II. III. 8. 3. 7. 2	Yes/no.
II. III. 8. 3. 7. 3	Yes/no.
II. III. 8. 3. 8	Yes/no, possibly a rating scheme.
II. III. 8. 4. 1	Yes/no.
II. III. 8. 4. 2	Yes/no.
II. III. 8. 4. 3. 1	Yes/no.
II. III. 8. 4. 3. 2	Yes/no.
II. III. 8. 4. 3. 3	Yes/no.
II. III. 8. 4. 3. 4	Yes/no.
II. III. 8. 4. 4	Yes/no.
II. III. 8. 4. 5	Yes/no.
II. III. 8. 4. 6	Yes/no.
II. III. 8. 4. 7	Yes/no.
II. III. 8. 4. 8	Yes/no.
II. III. 8. 5. 1	Yes/no.
II. III. 8. 5. 2	Yes/no.
II. III. 8. 5. 3	Yes/no.
II. III. 8. 5. 4	Yes/no.
II. III. 8. 5. 5	Yes/no.
II. III. 8. 5. 6	Yes/no.
II. III. 8. 5. 7	Yes/no.
II. III. 8. 5. 7. 1	Yes/no.
II. III. 8. 5. 7. 2	Yes/no.
II. III. 8. 5. 7. 3	Yes/no.
II. III. 8. 6. 1	Yes/no.
II. III. 8. 6. 2	Yes/no.
II. III. 8. 6. 3	Yes/no.
II. III. 8. 6. 4	Yes/no.
II. III. 8. 7	Yes/no.
II. III. 8. 7. 1	Yes/no.
II. III. 8. 7. 1. 1	Yes/no.
II. III. 8. 7. 1. 2	Yes/no.
II. III. 8. 7. 1. 3	Yes/no.
II. III. 8. 7. 1. 4. 1	Yes/no.
II. III. 8. 7. 1. 4. 2	Yes/no.
II. III. 8. 7. 1. 4. 3	Yes/no.
II. III. 8. 7. 1. 4. 4	Yes/no.

Reference Number		Measurement
II. III. 8. 7. 2. 1	Yes/no.	
II. III. 8. 7. 2. 2	Yes/no.	
II. III. 8. 7. 2. 3	Yes/no.	
II. III. 8. 7. 2. 4	Yes/no.	
II. III. 8. 7. 2. 5	Yes/no.	
II. IV. 1. 1	Yes/no.	
II. IV. 1. 2	Yes/no.	
II. IV. 1. 3	Yes/no.	
II. IV. 1. 4	Yes/no.	
II. IV. 1. 5. 1	Yes/no.	
II. IV. 1. 5. 2	Yes/no.	
II. IV. 1. 5. 3	Yes/no.	
II. IV. 2. 1. 1	Yes/no.	
II. IV. 2. 1. 2	Yes/no.	
II. IV. 2. 1. 3	Yes/no.	
II. IV. 2. 1. 4	Yes/no.	
II. IV. 2. 1. 5	Yes/no.	
II. IV. 2. 1. 6	Yes/no.	
II. IV. 2. 1. 7	Yes/no.	
II. IV. 2. 1. 8	Yes/no.	
II. IV. 2. 1. 9	Yes/no.	
II. IV. 2. 1. 9. 1	Yes/no.	
II. IV. 2. 1. 9. 2	Yes/no.	
II. IV. 2. 1. 9. 3	Yes/no.	
II. IV. 2. 1. 9. 4	Yes/no.	
II. IV. 2. 2	Yes/no.	
II. IV. 2. 2. 1	Yes/no.	
II. IV. 2. 2. 2	Yes/no.	
II. IV. 2. 2. 3	Yes/no.	
II. IV. 2. 3. 1. 1	Yes/no.	
II. IV. 2. 3. 1. 2	Yes/no.	
II. IV. 2. 3. 1. 3	Yes/no.	
II. IV. 2. 3. 1. 4	Yes/no.	
II. IV. 2. 3. 1. 5	Yes/no.	
II. IV. 2. 3. 1. 6	Yes/no.	
II. IV. 2. 3. 1. 7	Yes/no.	
II. IV. 3. 1. 1	Yes/no.	
II. IV. 3. 1. 2	Yes/no.	
II. IV. 3. 1. 3	Yes/no.	
II. IV. 3. 1. 4	Yes/no.	
II. IV. 3. 1. 5	Yes/no.	
II. IV. 3. 1. 6	Yes/no.	
II. IV. 3. 2. 1	Yes/no.	
II. IV. 3. 2. 2	Yes/no.	
II. IV. 3. 2. 3	Yes/no.	

Reference Number	Measurement
II. IV. 3. 2. 4	Yes/no.
II. IV. 3. 2. 5	Yes/no.
II. IV. 3. 2. 6	Yes/no.
II. IV. 3. 2. 7	Yes/no.
II. IV. 3. 2. 8	Yes/no.
II. IV. 3. 2. 9	Yes/no.
II. IV. 3. 3. 1	Yes/no.
II. IV. 3. 3. 2	Yes/no.
II. IV. 3. 3. 3	Yes/no.
II. IV. 3. 3. 4	Yes/no.
II. IV. 3. 5. 1	Yes/no.
II. IV. 3. 5. 2	Yes/no.
II. IV. 3. 5. 3	Yes/no.
II. IV. 3. 5. 4	Yes/no.
II. IV. 3. 5. 5	Yes/no.
II. IV. 3. 5. 6	Yes/no.
II. IV. 3. 6. 1	Yes/no.
II. IV. 3. 6. 2	Yes/no.
II. IV. 3. 6. 3	Yes/no.
II. IV. 3. 7. 1	Yes/no.
II. IV. 3. 7. 2	Yes/no.
II. IV. 3. 7. 3	Yes/no.
II. IV. 3. 7. 4	Yes/no.
II. IV. 3. 8. 1	Yes/no.
II. IV. 3. 8. 2	Yes/no.
II. IV. 3. 8. 3	Yes/no.
II. IV. 3. 8. 4	Yes/no.
II. IV. 3. 9. 1	Yes/no.
II. IV. 3. 9. 2	Yes/no.
II. IV. 3. 9. 3	Yes/no.
II. IV. 3. 9. 4	Yes/no.
II. IV. 3. 9. 5	Yes/no.
II. IV. 4. 1. 1	Yes/no.
II. IV. 4. 1. 2	Yes/no.
II. IV. 4. 1. 3	Yes/no.
II. IV. 4. 1. 4	Yes/no.
II. IV. 4. 2. 1	Yes/no.
II. IV. 4. 2. 2	Yes/no.
II. IV. 5. 1. 1	Yes/no.
II. IV. 5. 1. 2	Yes/no.
II. IV. 5. 1. 3	Yes/no.
II. IV. 5. 1. 4	Yes/no, possibly a rating scheme.
II. IV. 5. 2	Yes/no.
II. IV. 5. 3. 1	Yes/no.
II. IV. 5. 3. 2	Yes/no.

Reference Number	Measurement
II. IV. 5. 3. 3	Rating scheme.
II. IV. 5. 3. 4	Rating scheme.
II. IV. 5. 3. 5	Yes/no.
II. IV. 5. 3. 6	Yes/no.
II. IV. 5. 3. 7	Yes/no.
II. IV. 5. 3. 8	Yes/no.
II. IV. 5. 4. 1	Yes/no.
II. IV. 5. 4. 2	Yes/no.
II. IV. 5. 4. 3	Yes/no.
II. IV. 5. 5	Yes/no.
III. I. 1. 1	Yes/no.
III. I. 1. 2	Rating scheme.
III. I. 1. 3. 1	Yes/no.
III. I. 1. 3. 2	Yes/no.
III. I. 1. 3. 3	Yes/no.
III. I. 1. 3. 4	Yes/no.
III. I. 1. 3. 5	Yes/no.
III. I. 2. 1. 1. 1	Yes/no.
III. I. 2. 1. 1. 2	Yes/no.
III. I. 2. 1. 1. 3	Yes/no.
III. I. 2. 1. 1. 4	Yes/no.
III. I. 2. 1. 2. 1	Yes/no.
III. I. 2. 1. 2. 2	Yes/no.
III. I. 2. 1. 2. 3	Yes/no.
III. I. 2. 1. 3. 1	Yes/no.
III. I. 2. 1. 3. 2. 1	Yes/no.
III. I. 2. 1. 3. 2. 2. 1	Yes/no.
III. I. 2. 1. 3. 2. 2. 2	Yes/no.
III. I. 2. 1. 3. 2. 2. 3	Yes/no.
III. I. 2. 1. 3. 2. 3. 1	Yes/no.
III. I. 2. 1. 3. 2. 3. 2	Yes/no.
III. I. 2. 1. 3. 2. 3. 3	Yes/no.
III. I. 2. 1. 3. 2. 4	Yes/no.
III. I. 2. 1. 3. 2. 5	Yes/no.
III. I. 2. 1. 3. 2. 6	Yes/no.
III. I. 2. 1. 3. 2. 7	Yes/no.
III. I. 2. 1. 3. 2. 8	Yes/no.
III. I. 2. 1. 3. 2. 9	Yes/no.
III. I. 2. 1. 3. 2. 10	Yes/no.

Reference Number	Measurement
III. I. 2. 1. 3. 3	Yes/no.
III. I. 3. 1	Yes/no.
III. I. 3. 1. 1. 1	Yes/no.
III. I. 3. 1. 1. 2	Yes/no.
III. I. 3. 1. 1. 2. 1	Yes/no.
III. I. 3. 1. 1. 2. 2	Yes/no.
III. I. 3. 1. 1. 3	Yes/no.
III. I. 3. 1. 2	Yes/no.
III. I. 3. 1. 3	Yes/no.
III. I. 3. 2	Yes/no.
III. I. 3. 2. 1. 1	Yes/no.
III. I. 3. 2. 1. 2	Yes/no.
III. I. 3. 2. 1. 3	Yes/no.
III. I. 3. 2. 1. 4	Yes/no, possibly a rating scheme.
III. I. 4. 1	Yes/no.
III. I. 4. 2	Yes/no.
III. I. 4. 2. 1. 1. 1	Yes/no.
III. I. 4. 2. 1. 1. 2	Yes/no.
III. I. 4. 2. 1. 1. 3	Yes/no.
III. I. 4. 2. 1. 1. 4	Yes/no.
III. I. 4. 2. 1. 1. 5	Yes/no.
III. I. 4. 2. 1. 2. 1. 1	Yes/no.
III. I. 4. 2. 1. 2. 1. 2	Yes/no.
III. I. 4. 2. 1. 2. 1. 3	Yes/no.
III. I. 4. 2. 1. 2. 2. 1	Yes/no.
III. I. 4. 2. 1. 2. 2. 2	Yes/no.
III. I. 4. 2. 1. 2. 2. 3	Yes/no.
III. I. 4. 2. 1. 2. 2. 4	Yes/no.
III. I. 4. 2. 1. 2. 2. 5	Yes/no.
III. I. 4. 2. 1. 2. 2. 6	Yes/no.
III. I. 4. 2. 1. 2. 2. 7	Yes/no.
III. I. 4. 2. 2. 1	Yes/no.
III. I. 4. 2. 2. 1. 1	Yes/no.
III. I. 4. 2. 2. 1. 2	Yes/no.
III. I. 4. 2. 2. 1. 3	Yes/no.

Reference Number	Measurement
III. II. 1. 1	Yes/no, rating scheme.
III. II. 1. 2	Yes/no, rating scheme.
III. II. 1. 3	Yes/no.
III. II. 2. 1	Yes/no.
III. II. 2. 2	Yes/no.
III. II. 3. 1	Yes/no.
III. II. 3. 2	Yes/no.
III. II. 3. 3. 1	Yes/no.
III. II. 3. 3. 2	Yes/no.
III. II. 3. 4	Yes/no.
III. II. 3. 5	Yes/no.
III. II. 3. 6	Yes/no.
III. II. 4. 1	Yes/no.
III. II. 4. 2. 1	Yes/no.
III. II. 4. 2. 2	Yes/no.
III. II. 4. 2. 3	Yes/no.
III. II. 4. 2. 4	Yes/no.
III. II. 4. 3	Yes/no.
III. II. 4. 4	Yes/no.
III. II. 4. 5. 1	Yes/no.
III. II. 4. 5. 2. 1	Yes/no.
III. II. 4. 5. 2. 2	Yes/no.
III. II. 4. 5. 3	Yes/no.
III. II. 4. 5. 4	Yes/no.
III. II. 4. 6. 1	Yes/no.
III. II. 4. 6. 2	Yes/no.
III. II. 4. 6. 3	Yes/no.
III. II. 4. 6. 4	Yes/no.
III. II. 4. 7. 1	Yes/no.
III. II. 4. 7. 2	Yes/no.
III. II. 4. 7. 2. 1	Yes/no.
III. II. 4. 7. 2. 2	Yes/no.
III. II. 4. 7. 2. 3	Yes/no.
III. II. 4. 7. 3. 1	Yes/no.
III. II. 4. 7. 3. 2	Yes/no.
III. II. 4. 7. 4. 1	Yes/no.
III. II. 4. 7. 4. 2	Yes/no.
III. II. 4. 7. 4. 3	Yes/no.
III. II. 4. 7. 4. 4	Yes/no.
III. II. 4. 7. 5. 1	Yes/no.
III. II. 4. 7. 5. 2	Yes/no.
III. II. 4. 8. 1	Yes/no.
III. II. 4. 8. 2	Yes/no.
III. II. 4. 8. 3	Yes/no.
III. II. 4. 9. 1	Yes/no.
III. II. 4. 9. 2	Yes/no.

Reference Number	Measurement
III. II. 4. 9. 3	Yes/no.
III. II. 4. 9. 4. 1	Yes/no.
III. II. 4. 9. 4. 2	Yes/no.
III. III. 1. 1	Yes/no.
III. III. 1. 2	Yes/no.
III. III. 1. 3	Yes/no.
III. III. 1. 4	Yes/no.
III. III. 2	Yes/no.
III. III. 3	Yes/no.
III. III. 3. 1	Yes/no.
III. III. 3. 2	Yes/no.
III. III. 3. 3	Yes/no, possibly a rating scheme.
III. III. 4. 1	Yes/no.
III. III. 4. 2. 1	Yes/no.
III. III. 4. 2. 2	Yes/no.
III. III. 4. 3. 1	Yes/no.
III. III. 4. 3. 2	Yes/no.
III. III. 4. 4	Yes/no.
III. III. 4. 5	Yes/no.
III. IV. 1. 1	Yes/no.
III. IV. 1. 2	Yes/no.
III. IV. 1. 3	Yes/no.
III. IV. 1. 4	Yes/no.
III. IV. 2. 1	Yes/no.
III. IV. 2. 2	Yes/no.
III. IV. 3. 1	Yes/no.
III. IV. 3. 2	Yes/no.
III. IV. 4. 1	Yes/no.
III. IV. 4. 2	Yes/no.
III. IV. 4. 3	Yes/no.
III. IV. 4. 4	Yes/no.
III. IV. 5	Rating scheme.
III. IV. 6	Yes/no.
III. IV. 7	Yes/no.
III. IV. 8	Yes/no.
III. IV. 9. 1	Yes/no.
III. IV. 9. 2	Yes/no.
III. IV. 9. 3	Yes/no.
III. IV. 10. 1	Yes/no.
III. IV. 10. 2	Yes/no.
III. IV. 11. 1	Yes/no.
III. IV. 11. 2	Yes/no.
III. IV. 11. 3	Yes/no.
III. IV. 11. 4	Yes/no.

Reference Number	Measurement
III. IV. 12. 1	Yes/no.
III. IV. 12. 2	Yes/no.
III. IV. 12. 3	Yes/no.
III. IV. 12. 4	Yes/no.
IV. I. 1. 1	Yes/no.
IV. I. 1. 2	Yes/no.
IV. I. 1. 3	Yes/no.
IV. I. 1. 4	Yes/no.
IV. I. 2. 1	Yes/no.
IV. I. 2. 2	Yes/no.
IV. I. 2. 3	Yes/no.
IV. I. 2. 3. 1	Yes/no.
IV. I. 2. 4	Rating scheme.
IV. I. 3. 1	Yes/no.
IV. I. 3. 2	Yes/no.
IV. I. 3. 3	Yes/no.
IV. I. 3. 4	Yes/no.
IV. I. 4. 1	Yes/no.
IV. I. 4. 2	Yes/no.
IV. I. 4. 3. 1	Yes/no.
IV. I. 4. 3. 2	Yes/no.
IV. I. 4. 4	Yes/no.
IV. I. 5. 1	Yes/no.
IV. I. 5. 2	Yes/no.
IV. I. 5. 3	Yes/no.
IV. I. 5. 4	Yes/no.
IV. I. 6. 1	Yes/no.
IV. I. 6. 1. 1	Yes/no.
IV. I. 6. 2	Yes/no.
IV. I. 6. 2. 1	Yes/no.
IV. I. 6. 3	Yes/no.
IV. I. 6. 3. 1	Yes/no.
IV. I. 6. 4	Yes/no.
IV. I. 6. 4. 1	Yes/no.
IV. I. 7. 1. 1	Yes/no.
IV. I. 7. 1. 2	Yes/no.
IV. I. 7. 1. 3	Yes/no.
IV. I. 7. 1. 4	Yes/no.
IV. I. 7. 1. 5. 1	Yes/no.
IV. I. 7. 1. 5. 2	Yes/no.
IV. I. 7. 1. 5. 3	Yes/no.
IV. I. 7. 2	Yes/no.
IV. I. 7. 3. 1	Yes/no.
IV. I. 7. 3. 2	Yes/no.
IV. I. 7. 3. 2. 1	Rating scheme

Reference Number	Measurement
IV. I. 7. 3. 3. 1	Yes/no, rating scheme.
IV. I. 7. 3. 3. 2. 1	Yes/no.
IV. I. 7. 3. 3. 2. 2	Yes/no.
IV. I. 7. 3. 3. 2. 3	Rating scheme.
IV. I. 7. 3. 3. 3	Rating scheme.
IV. I. 7. 3. 4	Yes/no.
IV. I. 7. 3. 5	Yes/no.
IV. I. 8. 1. 1	Yes/no.
IV. I. 8. 1. 2	Yes/no.
IV. I. 8. 2	Yes/no.
IV. I. 8. 3. 1	Yes/no.
IV. I. 8. 3. 2	Yes/no.
IV. I. 8. 3. 3	Yes/no.
IV. I. 8. 4. 1	Yes/no.
IV. I. 8. 4. 2	Yes/no.
IV. I. 8. 4. 3	Yes/no.
IV. I. 8. 5. 1	Yes/no.
IV. I. 8. 5. 2	Yes/no.
IV. I. 8. 5. 3	Yes/no.
IV. I. 8. 5. 4	Yes/no.
IV. I. 9. 1	Yes/no.
IV. I. 9. 2. 1	Yes/no.
IV. I. 9. 2. 2	Yes/no.
IV. I. 9. 2. 3	Yes/no.
IV. I. 9. 2. 4	Yes/no.
IV. I. 9. 3. 1	Yes/no.
IV. I. 9. 3. 2	Yes/no.
IV. I. 9. 4	Yes/no.
IV. I. 9. 5	Yes/no.
IV. I. 9. 5. 1	Yes/no.
IV. I. 9. 5. 1. 1	Yes/no.
IV. I. 9. 5. 1. 2. 1	Yes/no.
IV. I. 9. 5. 1. 2. 2	Yes/no.
IV. I. 9. 5. 1. 2. 3	Yes/no.
IV. I. 9. 5. 2	Yes/no.
IV. I. 9. 5. 2. 1	Yes/no.
IV. I. 9. 5. 3	Yes/no.
IV. I. 9. 5. 4	Yes/no.
IV. I. 9. 6. 1. 1	Yes/no.
IV. I. 9. 6. 1. 2	Yes/no.
IV. I. 9. 6. 1. 3	Yes/no.
IV. I. 9. 6. 1. 4	Yes/no, rating scheme.
IV. I. 9. 6. 2	Yes/no.
IV. I. 9. 6. 3. 1	Yes/no.
IV. I. 9. 6. 3. 2	Yes/no.
IV. I. 9. 6. 3. 3	Yes/no.

Reference Number	Measurement
IV. I. 9. 6. 3. 4	Yes/no, rating scheme.
IV. I. 9. 6. 4	Yes/no.
IV. I. 9. 7. 1	Yes/no.
IV. I. 9. 7. 2	Yes/no.
IV. I. 9. 7. 2. 1	Yes/no.
IV. I. 9. 7. 2. 2	Yes/no.
IV. II. 1. 1	Yes/no.
IV. II. 1. 2. 1	Rating scheme.
IV. II. 1. 2. 2	Rating scheme.
IV. II. 1. 3	Rating scheme.
IV. II. 2. 1. 1. 1	Yes/no.
IV. II. 2. 1. 1. 2	Yes/no.
IV. II. 2. 1. 2. 1	Yes/no.
IV. II. 2. 1. 2. 2	Yes/no.
IV. II. 2. 1. 3. 1	Yes/no.
IV. II. 2. 1. 3. 2	Yes/no.
IV. II. 2. 1. 3. 3	Yes/no.
IV. II. 3. 1. 1	Rating scheme.
IV. II. 3. 1. 2	Rating scheme.
IV. II. 3. 1. 3	Rating scheme.
IV. II. 3. 2	Yes/no.
IV. II. 3. 3. 1	Yes/no.
IV. II. 3. 3. 2	Yes/no.
IV. II. 3. 4. 1	Yes/no.
IV. II. 3. 4. 2	Yes/no.
IV. II. 3. 4. 3	Yes/no.
IV. II. 3. 5. 1. 1	Yes/no.
IV. II. 3. 5. 1. 2. 1	Yes/no.
IV. II. 3. 5. 1. 2. 2	Yes/no.
IV. II. 3. 5. 1. 3. 1	Yes/no.
IV. II. 3. 5. 1. 3. 2	Yes/no.
IV. II. 3. 5. 1. 3. 3	Yes/no.
IV. II. 3. 5. 1. 3. 4	Yes/no.
IV. II. 3. 5. 1. 3. 4. 1	Yes/no.
IV. II. 3. 5. 1. 3. 4. 1. 1	Yes/no.
IV. II. 3. 5. 1. 3. 4. 1. 2	Yes/no.
IV. II. 3. 5. 1. 3. 4. 2	Yes/no.
IV. II. 3. 5. 1. 3. 4. 2. 1	Yes/no.
IV. II. 3. 5. 1. 3. 4. 2. 2	Yes/no.

Reference Number	Measurement
IV. II. 3. 5. 1. 3. 4. 3	Rating scheme.
IV. II. 3. 5. 1. 3. 4. 3. 1	Rating scheme.
IV. II. 3. 5. 1. 3. 4. 3. 2	Rating scheme.
IV. II. 3. 5. 1. 3. 4. 4	Rating scheme.
IV. II. 3. 5. 2. 1	Rating scheme.
IV. II. 3. 5. 2. 2. 1	Yes/no.
IV. II. 3. 5. 2. 2. 2	Yes/no.
IV. II. 3. 5. 2. 2. 3	Yes/no.
IV. II. 3. 5. 2. 3. 1	Yes/no.
IV. II. 3. 5. 2. 3. 2	Rating scheme, yes/no.
IV. II. 3. 5. 3. 1	Rating scheme, yes/no.
IV. II. 3. 5. 3. 2	Rating scheme, yes/no.
IV. II. 4. 1	Yes/no.
IV. II. 4. 2	Yes/no.
IV. II. 4. 3	Yes/no.
IV. III. 1. 1	Yes/no.
IV. III. 1. 2	Yes/no.
IV. III. 1. 2. 1	Yes/no, rating scheme.
IV. III. 1. 2. 2	Yes/no, rating scheme.
IV. III. 1. 2. 3	Yes/no
IV. III. 1. 2. 4	Yes/no.
IV. III. 1. 3	Yes/no.
IV. III. 2. 1. 1	Yes/no.
IV. III. 2. 2. 1	Yes/no.
IV. III. 2. 2. 2	Yes/no.
IV. III. 2. 3. 1. 1	Yes/no.
IV. III. 2. 3. 1. 2	Yes/no.
IV. III. 2. 3. 1. 3	Yes/no.
IV. III. 2. 3. 2. 1	Yes/no.
IV. III. 2. 3. 2. 2	Yes/no.
IV. III. 2. 3. 3. 1	Yes/no.
IV. III. 2. 3 3. 2	Yes/no.
IV. III. 2. 3. 3. 3	Yes/no.
IV. III. 2. 3. 4. 1	Yes/no.
IV. III. 2. 3. 4. 2	Yes/no.
IV. III. 2. 3. 4. 3	Yes/no.
IV. III. 3. 1. 1	Yes/no.
IV. III. 3. 1. 2	Yes/no.
IV. III. 3. 1. 3	Yes/no.
IV. III. 3. 1. 4	Yes/no.
IV. III. 3. 1. 5	Yes/no.

Reference Number	Measurement
IV. III. 3. 1. 6	Yes/no, rating scheme.
IV. III. 3. 1. 7	Yes/no.
IV. III. 3. 1. 8	Yes/no.
IV. III. 3. 1. 9	Yes/no.
IV. III. 3. 2. 1	Yes/no.
IV. III. 3. 2. 2	Yes/no.
IV. III. 4. 1. 1. 1	Yes/no.
IV. III. 4. 1. 1. 2	Yes/no.
IV. III. 4. 1. 1. 3	Rating scheme.
IV. III. 4. 2. 1	Rating scheme.
IV. III. 4. 2. 2	Yes/no.
IV. III. 4. 2. 3	Yes/no.
IV. III. 4. 2. 4	Yes/no.
IV. III. 4. 2. 5	Yes/no.
IV. III. 4. 2. 6	Yes/no.
IV. III. 4. 2. 7	Yes/no.
IV. III. 4. 2. 8	Yes/no.
IV. III. 4. 2. 9. 1	Yes/no.
IV. III. 4. 2. 9. 2	Yes/no.
IV. III. 4. 2. 9. 3	Yes/no.
IV. III. 4. 2. 9. 4	Yes/no.
IV. III. 4. 2. 9. 5	Yes/no.
IV. III. 4. 2. 9. 6	Yes/no.
IV. III. 4. 2. 9. 7	Yes/no.
IV. III. 4. 2. 9. 8	Yes/no.
IV. III. 4. 2. 10. 1	Yes/no.
IV. III. 5. 1. 1	Yes/no.
IV. III. 5. 1. 2	Yes/no.
IV. III. 5. 1. 3	Yes/no.
IV. III. 5. 2. 1	Yes/no.
IV. III. 5. 2. 2	Yes/no.
IV. III. 5. 2. 3	Yes/no.
IV. III. 5. 2. 4	Yes/no.
IV. III. 5. 3. 1	Yes/no.
IV. III. 5. 3. 2	Yes/no.
IV. III. 5. 3. 3	Yes/no.
IV. III. 5. 3. 4	Yes/no.
IV. III. 5. 4	Yes/no.
IV. III. 5. 5	Yes/no.
IV. III. 5. 6	Yes/no.
IV. III. 5. 7	Yes/no.
IV. III. 5. 8	Yes/no.
IV. III. 5. 9. 1	Yes/no.
IV. III. 5. 9. 2	Yes/no.
IV. III. 5. 10. 1	Yes/no.
IV. III. 5. 10. 2	Yes/no.

Reference Number	Measurement
IV. III. 5. 10. 3	Yes/no.
IV. III. 5. 11	Rating scheme.
IV. III. 6. 1	Yes/no.
IV. III. 6. 2	Yes/no.
IV. III. 6. 3	Yes/no.
IV. III. 6. 4	Yes/no.
IV. IV. 1. 1	Yes/no.
IV. IV. 1. 2	Yes/no.
IV. IV. 1. 3	Yes/no.
IV. IV. 1. 4	Yes/no.
IV. IV. 2. 1. 1	Yes/no.
IV. IV. 2. 1. 2	Yes/no.
IV. IV. 2. 1. 3	Yes/no.
IV. IV. 2. 2. 1	Yes/no.
IV. IV. 2. 2. 2	Yes/no.
IV. IV. 2. 2. 3	Yes/no.

SECTION V

DMS TEST SCENARIO

1. DMS EVALUATION RATIONALE

The problem of evaluating a DMS can be triggered by a variety of situations. Some examples of cases that might require the evaluation of a DMS are; 1) introducing a new piece of hardware into the environment (storage media), 2) improving the performance of a current DMS, 3) selecting a DMS for a given hardware environment, 4) selecting a DMS and a hardware configuration for a set of applications and 5) selecting a set of applications for a given DMS.

As might be expected, the answers to these problems do not stem from a universal technique but, dependent on the problem environment, a group of techniques can be merged to provide insight to a problem solution. Further, the group of techniques employed can vary widely from problem to problem.

Regardless of the techniques used, the evaluation of a DMS will usually encompass one or all of the following eight factors:

- Adaptability - can the system be adapted to new hardware or new applications.
- Cross-Referencing - does the system permit inter-and intra- file linkages.
- Data Compaction - does the system automatically edit data and compress record structures.
- Expense - are the initial implementation costs and operating costs justifiable.
- File Organization - does the system have flexibility in the types of storage structures and access methods used.
- Hardware Independence - can the system be easily transferred to another machine line or across generations.
- Tutoring Aids - does the system provide assistance to the user.
- User Interface - does the system provide necessary languages for the full spectrum of users.

Although there is no universal evaluation technique, but because these factors are part of every evaluation problem, a methodology for selecting an evaluation technique or techniques can be formulated. A methodology, in this context, being a set of systematic procedures that a DMS evaluation process would follow. The methodology hierarchy introduced in Section III fits this definition.

2. EVALUATION SELECTION GUIDELINE

The evaluation methodology alluded to in the above section would have flexibility as its most prominent characteristic. Not only flexibility in terms

of being able to address the many types of DMS evaluation problems, but also flexibility in terms of being multi-entrant and multi-exited. This latter type of flexibility is required to produce the former. For example, a DMS evaluation problem that addresses the selection of a DMS for a defined operating environment is going to envelop a DMS evaluation problem that addresses the selection of the most suitable file structure for a given application on an already selected DMS. The second evaluator will not tolerate going through the same procedures followed by the first evaluator and he should not have to tolerate them. The methodology should direct him to an entrance and exit point within the structure that solves his problem with the least amount of time and effort. The fact that each "DMS evaluation" brings with it a set of "givens" should be an advantage in the selection of evaluation techniques not a disadvantage.

Determining the methodology entrance point is not simply a process of tracing a path through the methodology until a point is reached where the "givens" no longer suffice. The nature of the DMS evaluation problem prohibits this simplistic an approach. However, the "givens" do provide an indication as to what level of the methodology hierarchy is of interest to the user. For example, the fact that a set of hardware requirements has been provided moves the user one level into the methodology but it does not select the evaluation technique a user would use to determine which CPU would best satisfy his DMS needs. In order to select the proper technique, the "givens" must be matched with the user's overall DMS needs.

3. EVALUATION SCHEDULING

Even when the "givens" and DMS requirements have been matched to select a DMS evaluation technique, the actual utilization of that technique is not guaranteed. Each technique, as shown in Section III, works best in a controlled environment. If this environment can not be provided or if it is disturbed, the data provided by the evaluation technique will be misleading or in some cases, erroneous. For example, running a set of benchmark programs to evaluate the retrieval capabilities of candidate DMSs requires that the benchmarks be designed to sample the actual retrieval types and task loads that will eventually be run under the selected DMS. If the benchmarks are poorly designed, the data they provide could bias not only the selection of the next evaluation step, but also the actual selection of the DMS.

If these environmental problems are to be avoided, consideration must be given to an implementation schedule for the entire evaluation task. This requirement means that a certain amount of analysis is going to be necessary in order to determine what the entire evaluation task will encompass. The user will have to know, within each level of the methodology, what techniques will be applicable. Each technique that is applicable will then be analyzed for the evaluation requirements it necessitates. If it is impossible to meet the requirements, other techniques at that level will be selected and the environment analysis process will repeat. In this manner, the user can describe the overall evaluation technique in terms of the testing environment. In addition, the scheduling of the evaluation task is accomplished and can be used as a guide to determine the status of the evaluation task.

4. EVALUATION OPERATIONS

The actual operations involved in the utilization of an evaluation technique are, in most cases, straightforward. This is especially true if the broad context of operations is being considered. For example, the use of a set of benchmark programs is relatively simple. The programs are designed to represent a sampling of the actual applications that will be run on the DMS; they are run; data is collected; and the data is analyzed. These basic operational steps, however, do not reflect the detailed and sometimes tedious analysis that often lies in the background. This analysis phase is virtually omni-present in these techniques, which besides benchmark programs, include software monitors, numerical scoring and any other analytical technique. When considering the schedule of an evaluation task, this analysis is always a prime consideration. Lack of it can often be the cause for poor data collection and eventually a poor evaluation.

While the operations of some evaluation techniques appear simplistic in a general context, others appear very complex. In this group, techniques such as modeling and simulation are included. Although "off-the-shelf" packages are available in these categories, it is very seldom that they are designed to give the user exactly what he requires. Even with limited modifications, these techniques require a knowledge of mathematics and/or Operations Research that the average evaluator seldom possesses. In many cases, this may cause the selection of other evaluation techniques but in still others, it causes the utilization of outside help. In any case, the actual operation of these techniques is much more sophisticated and this must be considered when an evaluation task plan is being developed.

5. EVALUATION DATA DISCRIPTION

The data collected from the use of a particular evaluation technique is similar regardless of the situation. This is true in spite of the many purposes and problems for which a technique is utilized. The interpretation and analysis of the data, however, varies from case to case and this, in turn, changes the results expected from the utilization of a technique. The following paragraphs describe the type of data collected when the named evaluation technique is utilized. No attempt is made to qualify the validity of the results.

a. Benchmarks

The data collected through the utilization of benchmark programs are timings. These timing can either be overall system timings or in the case of Kernel analysis CPU timings. Because these timings are not very refined and because the benchmarks are only a sample of the actual workload, they are usually coupled with other event timings and a ratio is used to decribe a certain phenomenon. For example, DMS 1 retrieval time is 31-1 better than DMS 2 for a given file structure but 1-26 for another type of file structure.

b. Numerical Scoring

This technique utilizes a weighted scoring approach. The user must first verify the DMS requirements and assign some numeric value as to the relative importance of the requirement in an overall DMS context. Each candidate is then judged on how well it satisfies the requirement and the score assigned is multiplied by the weight assigned to the requirement. The DMS awarded the highest points for that requirement would best satisfy that requirement and the DMS with highest overall point count would best satisfy the total DMS requirements.

c. Monitors

(1) Hardware Monitors

Usually a type of counter that records the occurrence of a significant event, i.e., Disc Access. The event is recorded outside the system operating environment and therefore does not interfere with the software. Two modes of counters are usually present in hardware monitors: a time mode and an incident or count mode. The count mode shows how many times a part of the system has been used in a given time period.

(2) Software Monitors

This type of monitor is embedded into the system software, is event dependent, and does affect the software operating environment. Timings are the main data collected by this technique and can be refined or as gross as the user wishes them to be. In most cases, the time is being recorded when a certain event occurs and recorded again once the event is over. The lapsed time serves as the measurement data.

d. Analysis

The most subjective of the techniques and therefore the most difficult to describe in regards to its results. The best that can be hoped for in this method is the knowledge that the system can not possibly provide the requirement being considered. A tool for elimination not evaluation.

e. Modeling/Simulation

The most difficult techniques to categorize as far as data collected is concerned. Because of the many levels of complexity that can be captured by these techniques, the collected data can range from timings, to ratios, to cost data. However, most of the data collected does revolve around refined system timings which may or may not be capable of being transposed to cost or value judgments.

6. EVALUATION VALIDATION

The techniques described as candidate DMS evaluation techniques, except for the analytic approach, provide some sort of quantifiable data as their output. Although this data can be related to some measure, usually time, it is

not necessarily meaningful. The final analysis step associated with any evaluation technique must prove the meaningfulness of the collected data. In this respect, the schedule of evaluation is most important. The evaluator must have some idea of what the collected data will be or its validity will have to be accepted. If the data is not what was expected, another technique could be used or, for that matter, the entire evaluation approach could be altered.

7. SAMPLE CASES

The previous six sections of this paper have described some of the concepts that must be considered in any DMS evaluation. In order to further amplify these comments and give some insight into the use of the methodology hierarchy, four DMS evaluation case studies are presented in this section. The cases represent a cross-section of the evaluation problems and specifically address:

- o Determining how to improve the responsiveness of a current DMS
- o Lessening the storage requirements for multi-user files under a current DMS
- o Selecting a DMS for a given host environment.
- o Selection of a DMS/hardware configuration.

Each case study will consist of three parts: the situation, the evaluation process, and the summary.

a. Case I

(1) The Situation

A set of DMS users in the XYZ corporation have become aware of problems concerning the responsiveness of their in-house DMS in both the query and update modes. Since the DMS was a relatively major investment in terms of analysis time and money, it was decided to attempt to improve the user interface with the DMS rather than replace it. At this point, an evaluation plan was formulated. This plan consisted of a review of the initial major goals of the DMS and based on these goals an evaluation of how certain users were or were not achieving the goals.

(2) The Evaluation Process

Prior to selecting the current DMS, three major goals or objectives were outlined. These can be summarized as follows:

- o Greater Data Independence. Two areas of independence were identified, the first was between data and programs, and the second between data and storage devices. That is, if data definitions are changed, programs do not necessarily have to change and further, if storage devices are changed, data definitions and programs remain relatively unchanged.

- o Content Retrieval. It was desired to retrieve data records on the data contents of the records, and not just on the key upon which the records are sequenced. Instead of asking "Retrieve records for employee 1234", it is desired to ask "Retrieve all records for employees who are between the ages of 25 and 35, who have a master's degree in engineering and who speak French". With records in mass storage, it is possible to find all of the desired records without having to search the entire file.

- o Fast Response. Another desirable feature was the ability to answer queries and to update the file rapidly. The problems associated with answering queries rapidly pose evident questions about file design. This is particularly true when the queries are complex and content retrieval is required. By the same token, the problems associated with file updating may not appear serious until it is realized that updating includes the insertion of new records and the lengthening of existing records as well as finding the records. With these objectives in mind, the evaluation criteria used to select the current DMS was heavily weighted in favor of a DMS that offered good file organization and management techniques. Therefore, the prime identifiers of the current DMS are:

  - Rapid and immediate classification of incoming data
  - Quick accessibility to the desired fraction of the present data base
  - A reservoir of up-to-the-second-data

  Both the query and update capability of the current DMS rated as the "best" available. However, these two areas, queries and updating, have at one time or another posed problems to all the DMS users. For example, a survey conducted prior to the evaluation revealed that delays of several minutes had been realized for queries, and extensive file updates had caused entire files to be removed from the system.

Based on this data, the first step in the evaluation process was to have each DMS user provide a set of benchmark programs that would best represent his DMS workload. Because these programs would be run on the DMS, it was felt that they would provide a meaningful way of determining the running times and resources required by different users. It was also felt that the results from the benchmarks would provide the evaluators with not only the processing time, but also the throughput time for specific applications without wading through the entire DMS workload. The results from the benchmark run, however, were only as good as the programs themselves. In this case, the benchmarks were inefficient and poorly conceived and, therefore, the results were not a true measure of the system. With these problems, it was also possible that the benchmarks may have operated "against" the

system by not utilizing the truly important capabilities of the system or by exaggerating its faults.

For the above reasons, the benchmark program data was given little weight and other evaluation techniques were considered. However, one thing the benchmarks did confirm was that complex queries resulted in long response times, and that long file updates produced irrecoverable system errors which often resulted in the loss of an entire file. Also, the benchmarks indicated various areas that might be candidates for investigation as system trouble spots. Note, that in this case, benchmarks were used as an indicator for trouble spots, not as a pure evaluation tool. Based on this, a set of software monitors were embedded in the DMS software to collect timing data on the file generation, retrieval, and update capabilities of the system. The timings were based on system clock time. Each file generation, retrieval or update was assigned a unique event number and within each event a discrete set of sub-events was also monitored. For example, a file retrieval event was broken down into 1) determining what record type held the requested information, 2) accessing an index for the value, 3) building a hit list of record pointers or a sequential search of a file, 4) sorting the hit list, and 5) retrieving records based on the hit list pointers. The start and stop for each event, and within each event, for each sub-event, was recorded on a journal tape. Later, a reduction program interpreted the data collected on the journal tape and produced a report with the data for each event and its sub-events grouped together.

The monitors were left embedded in the system for several weeks and data was collected for the system operating under almost all conditions. When the results were compiled and analyzed, it was evident to the evaluators that the structure defined for a given file was directly linked to the amount of time required for a given retrieval or update task. For example, a query which initiated a large retrieval on an indexed element of a file was much faster than a retrieval initiated on an item that was serially arranged. This data seemed to indicate a solution to the retrieval problem. However, the building of the index for the element has an overhead cost that is not reflected in the retrieval process and in order to evaluate one file structure against another, this and other factors must be considered. With this in mind, tasks were initiated to create indexes for various file elements and queries were performed for the elements. The software monitors recorded the time required, and when the data was reduced and correlated, an interpretation process followed.

The interpretation of the collected data showed that most of the problems isolated by the software monitors were directly linked to the file structure selected by the user and then, based on this structure, the type of access method he chose to utilize. For example, some queries which resulted in an output of only a few records where satisfied only after a large number of accesses were initiated. This was caused by either a poor indexing scheme (an indexed file structure), a poor decoding algorithm (a randomly structured file), or accessing a serially arranged file. Also, short update tasks were consuming too much time. This problem was also related to the type of file structure utilized and in most cases reflected a poor sort order on a serially arranged file.

Based on the findings of the interpretation task, two further evaluation steps were selected. The first was to analytically examine the DMS software documentation to determine if there were any limitations to the file organization and accessing capability of the DMS. The second was to model the organization and accessing environment and then determine the best techniques for a given application.

The investigation revealed that the DMS allowed either a hierarchical or a single level file structure. Within these two structures, a number of access methods were available. These methods were linked directly to the IBM 360 hardware on which the DMS was run and included BSAM, QSAM, BDAM, and BISAM. By modeling the 360 hardware characteristics, operating system and access methods, the evaluators were able to examine the behavior of different file organizations and access methods within the environment and usage patterns that had been previously determined. Each user's file strategies and usage patterns were then varied and the model was used to predict the outcome of the changes. In this manner, the evaluators were able to determine the best file organization, hardware utilization and access method for each user. It also provided a technique to aid users in determining the same requirements for new applications or hardware.

(3) Summary

Although the route to the solution of the DMS evaluation problem may seem circuitous, the steps taken were logical and within the context of the methodology hierarchy. The set of givens provided may have caused an evaluation to initially skip the use of the benchmarks, but the analysis leading to the placement of the software monitors would undoubtedly lead him back to the benchmark step. As with the benchmarks, the software monitors did not provide a solution to the problem, but again they did point the evaluator to the next step. The ability to rapidly diagnosis this type of situation depends on the amount of scheduling and background analysis performed prior to the actual evaluation. Note that the overall process eventually provided not only a technique which solved the present problem, but also could provide payoffs if future DMS problem areas arose.

b. Case 2

(1) Situation

A large branch of the military has several data processing installations of similar configuration. While each installation has many local files used in day-to-day processing, it has been determined that several sites are maintaining duplicate files which are used with varying frequency. Furthermore, the mode of use of these shared files varies; in some cases they are just maintained, other sites primarily perform read operations, while still others require both read and update access. It is obvious that there are various unnecessary storage costs associated with maintaining multiple copies of files. Can these be eliminated by having only one copy of each file? Heuristically, each copy would be located at the installation where it is most used - but again, "use" can be for reading, writing, or both so this allocation may not be a good rule to enforce. With single copies of course, there would be

transmission costs involved in allowing telecommunications access by other sites on request. Hopefully, any transmission cost would be outweighed by the savings realized in eliminating file residence at multiple locations.

A model of such a system can be developed which allows the evaluation of the cost savings obtained by interconnecting the computer installation with the necessary transmission links and allocating the files properly among the nodes of the network. It can then be determined whether such a system is worthwhile.

(2) The Evaluation Process

The problem as abstracted for the mathematical model is one in "distributed" data bases. The situation considered is one in which there are geographically separated but completely interconnected computer systems containing multiple shared files. A method for arriving at optimum performance in terms of operating costs is desired. The variables of concern are capacity and cost of file storage at each site, the rates of file modification, transmission, and access requests, and the line capacity for transmission. Cost is defined as transmission cost plus storage cost. The approach is to formulate the problem as a zero-one integer programming model where:

$$X_{ij} = \begin{cases} 1 \text{ if file } j \text{ is stored at } i\text{'th computer}, & i=1,2,\ldots,n \\ 0 \text{ if it is not} & j=1,2,\ldots,m \end{cases}$$

In words, given n computers for processing m distinct and common information files, how can one allocate the files so that a minimum overall operating cost is achieved under the following constraints: (a) expected file access time is less than some upper bound; (b) the storage needed at each site does not exceed capacity.

For n computers, simultaneous transmission in both directions is assumed to occur on a pair of paths - one for requests and one for replies for each computer (single channel, full duplex).

The cost C is given as a linear combination (the objective function) of the factors which comprise storage cost and transmission cost. (The cost of <u>installing</u> the transmission links is not considered.) These factors are:

c_{ij} = storage cost for file j at i per unit length

c'_{ik} = transmission cost from k to i per unit time

U_{ij} = request rate for all or part of j by i per unit time. For actually exercising the model, an assumption must be made as to the behavior of the file access requests. A widely used assumption is that these events behave statistically with a Poisson distribution.

P_{ij} = modification frequency of j at i

t_j = transmission time for j (based on line capacity)

L_j = size (length) of j'th file.

It is noted that when only one copy of a file exists among the computers, $X_{ij}X_{kj} = 0$, $i \neq k$; the problem is linear and can be solved by standard linear programming techniques. The constraints under which the cost C must be minimized are:

1. $\sum_i X_{ij} L_j \leq b_i$ — for all i (Storage Capacity not exceeded)
   b_i = availability capacity of i'th computer
   L_j - defined above

2. $X_{kj} a_{ijk} \leq T_{ij}$ — for all j, $i \neq k$ (File Access time acceptable)
   T_{ij} = maximum allowable retrieval time of j from i
   a_{ijk} = time to get file j from k to i

If it is desirable to allow more than one copy of a file at different computers (i.e., "redundancies," $\sum_i X_{ij} > 1$ for any j), the problem is no longer linear and standard techniques cannot be used. However, it may be possible to devise additional constraints which will, in effect, "linearize" the problem for standard solution.

Once these steps have been completed, input parameter variations are devised to specify the simulation experiments. The data collected from the experiments can then be validated to determine the feasibility of devising such a distributed data base system.

(3) Summary

The use of the model/simulation as an evaluation technique is not always as obvious as it was in this "textbook" case. In selecting this technique, the availability of an evaluator with a mathematical background is almost a necessity if the process to be correctly developed and the results are to be correctly analyzed. In the process of setting up the evaluation guidelines, the net worth of this type of an approach should be very apparent and the decision to proceed or attempt another technique should be made immediately.

c. Case 3

(1) Situation

Corporation DEF is a manufacturer of heavy equipment in the Eastern and Central section of the United States. The company has several plants, each specializing in a particular type of heavy equipment such as ship building, aircraft parts, and farm machinery. In the early 1960's, the farm machinery plant initiated a computerized Production Planning System (PPS). This system proved to be so successful that corporate headquarters decided to purchase third generation hardware, centralize the plant EDP

operations at corporate headquarters and have all the plants adopt the Production Planning System.

The primary motivator behind this decision was headquarter's desire to enter new and diverse marketing areas. The PPS had cut production planning lead time at the farm machinery plant and it was hoped that the incorporation of these same computer based techniques would do likewise for new marketing targets. However, the data or files manipulated by the dedicated PPS at the farm machinery plant were easy to define and not overwhelming in size. On the other hand, a corporate undertaking along these lines necessitated the merging and management of not only copious amounts of data but also diverse data. In order to make the PPS workable on a corporate level, it was decided that a Generalized Data Management System would be required.

(2) The Evaluation Process

Once the requirement for a GDMS was arrived at, the process of selecting the right one began. The first step in the selection was to decide whether a given GDMS would be capable of processing the required applications. In addition, this step uncovered any and all requirements for the GDMS, as well as provided the givens necessary to enter the methodology hierarchy. Two kinds of requirements: mandatory and desired were investigated. The distinction between the two categories is as follows: mandatory items are those that are essential to the implementation of the company's needs, while desirable items are those which would make the implementation of the company's needs easier. Within this structure, all systems that provided the mandatory requirements would be considered and any desirable items would make one system more or less advantageous when compared to another.

Most items considered under the heading of mandatory requirements are concerned with one of the following aspects of data management systems:

- Cost
- Due dates - implementation date
- Application capabilities - update, retrieval
- Language - query, application
- Device/security interface
- Hardware configuration/operating environment
- Operating Systems
- Input/output requirements

Once the mandatory and desired features of the system have been determined, they can then be applied to the list of candidate DMSs to determine which systems qualify. When all candidates have been screened in this manner, the qualified candidates can then be evaluated to determine the "best" system.

At this juncture, a list of acceptable DMSs has been compiled but more important, the process used to determine this list should have given the evaluators a thorough knowledge of user requirements. These requirements can now be translated into general DMS functions. This general background coupled with DMS systems documentation and a numerical scoring method can now be used to rate the candidate DMSs.

In a numerical weighing technique, the parameters of the required DMS are weighted according to their priority in the operational environment and then compared to the capabilities of each candidate DMS and a rating assigned. Although this process is manual and in some cases becomes a purely subjective analysis tool, it can serve two important purposes. First, it further defines the criteria against which candidate DMSs are evaluated, and secondly, it provides a ranking of the candidate systems from which the poorest can be eliminated. But, since the numerical scoring is based on system documentation, which can be misleading, and because the rator can not help but be subjective, this methodology was used only as one of the first steps in the overall evaluation process.

The next logical step in the evaluation process was to run benchmark programs on the remaining candidate systems but because several were still in contention, the cost and time involved was judged to be too expensive. Instead, an intermediate analysis step was performed. The objective of this step was to collect and evaluate actual operational statistics for the candidate systems. During the course of the task, numerous candidate system users, as well as programmers and analysts that interfaced with the DMSs were interviewed. Since these interviews could be misleading, the DMS was observed running whenever possible. Not all data collected in this manner could be used. For example, in some cases, system problems were observed and they could not be directly linked to the operating system, DMS or application programs, but interviewees were quick to attribute these problems to DMS software. By the same token, the interviewees could be extremely positive about DMS capabilities and attribute OS or application program advantages to the DMS. Regardless of its obvious shortcomings, this evaluation step gave the evaluators a feel for how a given DMS actually operated. It moved the evaluation process from the theoretical basis of documentation to the real operational world. Any blatant operational problems were easily spotted and any advantages were also obvious, and the cost to determine the results was minimal. When the analysis step was completed, only the most adequate DMS candidates remained.

With the list of original candidates reduced to just the best possible candidates, Corporation DEF had to decide what final evaluation technique would be used to make the final selection. Two techniques were considered by the evaluation group: simulation and benchmark programs. In deciding between the two techniques, the evaluation group considered the cost of the technique, its responsiveness, and the meaningfulness of the generated output. Because more than one DMS would be evaluated, the cost for either technique was very high. This was still true after the DMS development companies offered to make operating systems available for testing purposes. In the end, since no simulations of the system were developed and the evaluators

felt a good set of test programs could be developed, the benchmark evaluation technique was selected.

Arriving at the mix or grouping of applications to be executed on the candidate DMSs was relatively simple. Since PPS was operational, the programs just had to be converted to run under the DMS. However, a problem did arise in trying to develop a standardized operating environment in which to run the tests. Once this was accomplished, representative programs could then be run on the proposed DMS and the measurements could be compared for both throughput and processor advantages. Because the environment had been standardized and DMS-OS interface problems could be considered constant, the entire system, not just the DMS, could be evaluated in the same process. The final results could then be evaluated and the "best" DMS selected.

(3) Summary

Selecting a DMS for a given set of applications can, in many ways, be much easier than trying to alleviate problems that are hampering a currently owned DMS. Although the techniques used are similar, the criteria the user is evaluating is much more general and in some cases is directed at a process of elimination rather than selection. The process observed in the Case 3 scenario supports this viewpoint. In this example, the previous selection of hardware and a detailed set of application programs narrowed the field of candidate DMSs, but it still allowed the evaluators a lot of flexibility in arriving at a methodology entrance point and eventually the selection of a DMS.

Documentation is a valuable source of data in this type of evaluation process. It is the prime source for the numerical analysis methods and it is used extensively in the analysis phase. However, documentation can only provide an initial insight into the DMSs capability and at that it is often subject to gross mis-information. Because of this, any DMS that is not eliminated during the analytical phases of evaluation should, if possible, be viewed in an operational environment before any other evaluation techniques are implemented. The actual analysis of an operating DMS can often answer many of the evaluators questions on OS/DMS and DMS/user interfaces. Only after these two areas are operationally evaluated should a simulation or benchmark evaluation be performed to determine the final DMS selection. The technique used in the final phase of the process will vary from evaluation to evaluation and will depend on 1) the cost of the technique, 2) the availability of the technique, and 3) the time required to utilize the technique.

d. Case 4

(1) Situation

The PDQ corporation has just authorized the creation of a centralized corporate management information system (MIS). Until this authorization was approved, ten separate and unequal MISs existed within the corporation. Each MIS served one of the nine branches of the corporation with the tenth serving corporate headquarters. Since the nine branches operated in functional areas quite unfamiliar to the others, the data manipulated as well

as the output generated by the MISs was very dissimilar. Added to this application dissimilarity was the multitude of different hardware on which each branch installed its MIS. The only commonality between the ten operating MISs was that all the application programs, although dissimilar, were written in COBOL. Finally and most importantly, the data received by the headquarter's MIS was quite limited and general in nature. Corporate management, therefore, felt the system was useless and returned to manual methods to support the decision making processes. This, however, also proved to be unsatisfactory because of the excessive amount of resources required to collect, prepare and disseminate the necessary reports. Corporate management thereupon decided to establish a centralized corporate MIS and authorized the performance of a requirements analysis.

The consolidation of this maze of programs, data and applications under one centralized system was a gigantic task. In order to solve it, a two phase approach was initiated. The first phase of the plan called for the selection of a hardware configuration to support the centralized system. A new hardware configuration was required because none of the present hardware configurations were large enough in terms of disc storage, number of peripherals, core size and speed to support the entire system, and since each branch had purchased hardware from different vendors, it was highly unlikely that several systems could be merged to support the new MIS. Further, the current systems still could be used by the branches for in-house accounting as well as production control.

The second phase of the MIS Implementation Plan called for the evaluation and selection of a generalized Data Management Syste[illegible] The DMS was to be the software link between the maze of data input b[illegible] nine branches into the corporate data base. The system selected would be run under the new hardware configuration and together with the har[illegible]are would be the basis for the new MIS.

Perhaps the selection group's mo[illegible] [illegible]portant decision was made prior to seeing any hardware or software. Th[illegible] [illegible]cision was that the DMS and hardware configuration eventually selected would not necessarily be the best in their respective categories, but when combined, would possess the highest rating. Because of this decision, the two phases of the selection would become intertwined and utlimately lead to the selection of best combination of hardware and software.

(2) The Evaluation Process

The first step in the testing process was the determination of the applications requirements of the corporation. The requirements study, if done well, would translate into EDP terminology, the reason why the system is ne[illegible]ded, identify the system users and define the user's information needs. The three most important results of this step in regards to the selection process should be: 1) a statement of the system objectives, 2) a list of environmental features most likely to affect the scope of the system, and 3) a list of the restriction that bear upon the scope of the system. Although most of the results of this step have already been discussed in

previous meetings, etc., the requirements study represented the first time they have been made official and hopefully validated. It is here that the entire project received the corporate seal of approval. Any hostility or lack of cooperation for the project should disappear here.

The applications requirements analysis indicated that a third generation multi-programming computer system was required with sufficient speed, core size and direct access storage to handle the processing load that would be imposed on it by the MIS. An on-line capability linking via a communications network the nine branches with the centralized system also was needed. The nine branches would supply on-line and batch updates to the data file which would periodically be queried by corporate management for status reports. The MIS also would be required to produce the various corporate reports on personnel, sales, profit-loss, etc., that are needed by management. The capability to produce these reports in hard copy or have them displayed on a CRT also would be required. Additional requirements, of a batch nature, would be to generate the monthly statements, print the employees' checks and handle the accounts payable and receivable.

Once these requirements were validated, it was obvious that the present system or a combination thereof could not handle the processing load. Therefore the hardware selection process began. Within this phase, several steps exist. These include a review of the specifications of the system, a determination of the evaluation techniques to be used, selection of the procurement method, development of validation techniques and determining the most prudent way to deal with vendors.

The starting point of any plan for selecting a computer system should be the review of the specifications of the system, since it is these specifications that define what it is that the corporation is seeking in the way of a computer system. These specifications reflect the findings of the requirements step which analyzed the corporate MIS needs. Before the specifications are applied to any candidate hardware configuration, they must be translated into mandatory and desirable features. The mandatory features are those items essential to the implementation of the system and the desirable features are those items which would make the implementation of the corporate needs easier. In this particular case, the mandatory features are a particular core size, processing speed and a random access storage and teleprocessing capability. The system must be able to perform on-line and batch operations, and be capable of supporting a qualified generalized data management system. Low cost, compatibility, responsiveness, and reliability also were required. The desirable system features were concerned with such hardware features as automatic interrupt, floating-point arithmetic, memory protections and indirect addressing and such software features as the compilers supported and operating system capabilities.

Once the differentiation between mandatory and desirable features of a system has been made, the mandatory features offer the first measure of the degree to which a vendor has succeeded in meeting the user's requirements. Either a vendor satisfies these requirements or his proposal is no longer considered for evaluation. However, it can be expected that more

than one vendor will satisfy the mandatory requirements. The purpose of the next step in the selection process, the evaluation, is to find the system that best satifies the corporate needs. Several techniques have been developed for hardware evaluation. These can be listed as follows:

1- Sole Source - staying with one vendor because of prior service, etc.

2- Overall Impression - a subjective judgment on the written or verbal proposals given by vendors. Controlled by "human nature."

3- Cost Only - Since all the systems being evaluated meet the mandatory requirements, buythe cheapest one.

4- Weighted Scoring - The user assigns points to all items he considers important and then selects the one with the most points. With this technique, it is almost impossible to arrive at a point relationship between low cost and high performance.

5- Cost/Effectiveness Ratio - Similar to Weighted Scoring, except that here, by dividing the cost by the effectiveness of the system, the lowest cost for effectiveness can be found. However, it is still questionable if a meaningful relationship can be established between these two factors, cost and effectiveness, on an overall system basis.

6- Cost-Value technique - Since all systems provide the mandatory features, this technique concentrates on evaluating the desirable features and validating the mandatory features. Simply stated, this technique subtracts the cost-value of the desirable items from the total cost of the proposed system. The difference is then considered to represent the derived cost of satisfying the requirements.

Regardless of the approach selected, some sort of rating system, perhaps similar to PEGS, can be established for those configurations which satisfy the mandatory requirements. When this ranking is compiled, the second phase of the selection process, testing of the DMSs, can commence.

The first step in this process is the conversion of applications requirements to specific DMS requirements. This process yielded the following DMS requirements:

- integrated data base structure
- direct access methods
- powerful procedural language
- report generation capabilities

- o encoding/decoding functions
- o powerful file generation and maintenance capability.

Additionally, the selection of a DMS for a hardware configuration not yet selected can pose several problems unique to DMS evaluation. Perhaps the most important of these is the evaluation of self-contained versus host data management systems. A host language system is one which is embedded in a procedural language such as COBOL or PL/1. A host system can best be visualized as a new tool for the application programmer. In addition, the DMS facilities can in some cases, be used by the assembly language programmer. A self-contained system, on the other hand, offers a tool for the nonprogrammer as well as the programmer. Such systems are self-contained in that they have no connection with a procedural language.

The differences between the two classes of DMSs are embodied in more than just the use of a procedural language. For example, a host system allows much more flexibility in the use of the system since unique programs can be written in the procedural language with the DMS acting as a data structuring or transferring agent whereas self-contained systems are dependent on MACROS which trigger "canned" routines. On the other hand, self-contained systems appeal to a larger set of users in that no knowledge of a programming language is required. This last attribute allows several levels of DMS users, but it restricts the more sophisticated user to a pre-determined sequence of events. For this reason, a self-contained system is detached from the user. The user has little or no feeling for the subtleties of storage structures or file updating and interrogation strategies. If a decision can be made between these two broad groups of DMSs, the entire phase 2 of the selection is cut in half. The most important characteristic to consider in making this decision is the level of the user. Since this factor was determined in the requirements study for phase 1 of the selection, the class of DMSs to consider should be readily apparent.

Now that the DMS criteria have been determined, the actual testing of the candidate systems can begin. The first step stipulates that documentation be gathered on the available data management systems, after which each system will be reviewed to determine if it possesses the MIS required capabilities. As part of this analysis, various installations within the area that utilize any of the candidate DMSs were visited and their operations investigated. The test personnel attempted to find a user with a similar application to discuss in depth the performance of the DMS. This was possible, although only one DMS in such an environment could be observed.

The analysis effort identified three DMSs as potential candidates for selection. Because only three systems surfaced from the analysis effort, the use of the numerical scoring technique, which would have been the next technique utilized, was bypassed. Also, the user applications did not possess the diversity to warrant utilization of numerical scoring.

Therefore, the testing was ready to proceed with the utilization of active techniques. The first active level in the methodology suggests that benchmark programs and/or hardware monitors be employed to derive overall

system performance. Because the user possessed neither a hardware monitor nor the expertise to use one, it was quickly decided that benchmark programs should be used. The test personnel hoped that they would not be required to code an extensive number of benchmarks but that the DMS designers would provide some that closely resembled their application requirements, or that they could convert some of their own applications software to serve as benchmarks. This latter hope was facilitated to some degree by the fact that all the present applications programs were written in COBOL. These programs, however, did not provide the on-line query and update capability required. Additionally, the data would have to be reorganized into an integrated data base structure against which the benchmark software would execute. The DMS designers did possess some "standard" benchmarks, but these were not typical enough to provide a valid test, therefore test personnel generated some normal type applications benchmarks to test the three systems. Four benchmark programs were written. The first one would test the file generation capabilities of the DMS, the second would measure the maintenance capabilities, the third would test the retrieval speed based on standard type queries while the fourth would specifically measure the formatting and output capabilities of the DMS. Properly coded, the benchmarks should test approximately 75 percent of the DMS capabilities, the remaining 25 percent being certain access methods that would not be appropriate to the particular application, specific retrieval expressions deemed inappropriate and other general system features that test personnel felt would seldom, if ever, be used by the corporate system.

The benchmarks were to be written in COBOL and it was hoped that only minor modifications would be required to permit their execution under each DMS.

At this stage of the testing, the possible hardware systems that could be selected also would be narrowed based on the remaining DMS candidates. If the DMSs are truly machine-independent then the hardware selection can be based solely on machine performance. However, if a host DMS is a candidate, then the particular machine on which the DMS runs must be an integral part of the test process.

Thus, in the selection process, the two phases (hardware and DMS selection) have become one, and the set of benchmarks was to run under all configuration combinations. Approximately twelve runs per system were required due to system aborts and to provide a mean time for each benchmark.

The generation of the benchmarks consumed a significant amount of human resources in addition to the computer resources utilized when the tests were actually run. The result of these tests was a series of timings indicating the amount of elapsed time to build a file and data base, maintain it, retrieve from it and output segments of it in a formatted mode. Not only were total system timings provided, but the test personnel developed a "feel" for each system; including both its strong and weak points.

At the conclusion of benchmark testing, no system was clearly superior to the others in terms of overall system performance. Corporate

management, therefore, decided to conduct testing at a more detailed level and, thus, descend deeper into the hierarchy. The techniques available were kernel analysis, software monitoring and modeling.

Kernel analysis and modeling both were eliminated from consideration; the former because the necessary accounting routines did not exist on all systems to collect the required timings while the latter technique was too costly to develop in terms of human resources, since none of the test personnel were familiar with model generation. Software monitoring, therefore, was selected.

At first it was hoped, that the DMSs under consideration had monitors already embedded into their modules. This, however, proved to be a false hope, therefore the test personnel began the task of analyzing the documentation of each DMS to identify the modules most appropriate for monitorization. They were fortunate to the extent that good documentation with detailed system flow charts was available on all DMSs. Where possible, corresponding modules from each system were selected. The modules chosen concerned file generation, data retrieval and data maintenance and plugs were implanted in each module. Of special concern were those routines that performed address calculations, built and searched directories and/or indices and generated data linkages. The implanted plugs would call a specialized subroutine which would calculate the time expended during each execution of the module. A record containing this information, then, would be created and written onto magnetic tape.

Subsequent to the modifications of the DMSs to include these plugs and the timing subroutine, the benchmark programs run previously were re-executed. Approximately the same number of executions as required for benchmark testing were required to collect a reasonable sample of performance statistics generated by the software monitors. Additionally, total system elapsed time was also calculated to determine the decrease, if any, in system efficiency caused by the embedded monitors.

The test personnel knew the volume of data that would be collected by the software monitors, therefore, they also wrote a data reduction program to process the output tape housing the various timing data. This program organized the timings both sequentially and by event type to ease to some degree the subsequent analysis. The timing data collected on each system were applied to the following questions:

1- How efficient is the system being tested? Are its resources being over or under utilized?

2- Does the DMS degrade overall system efficiency?

3- Are there any "bottlenecks" in the system?

4- What are the causes of "wait" states in the system? Are they excessive?

5- Can the system serve the level of user previously decided on? Is it responsive enough?

6- What is the performance rate of each module vis-a-vis the corresponding modules in the other systems?

Extensive analysis was required to answer these questions and consideration was given to using an analysis technique such as regression and/or cluster analysis to determine system performance and aid test personnel in results evaluation. These techniques, however, had only been used in analyzing total system performance and not specific software subsystems such as a DMS. Therefore, the various dependent and independent input variables had not been identified to permit the utilization of such techniques. Test personnel, therefore, manually accomplished the results analysis which indicated that one system (including both hardware and DMS) was indeed most appropriate for their particular applications. This system was then recommended to corporate management which approved its selection.

The other techniques in the hierarchy, combining hardware and software monitors and using simulation/modeling, were therefore not required and the testing ended.

(3) Summary

The selection of a hardware configuration and a DMS to operate on that configuration can best be approached in two phases. The first phase is dedicated to evaluating various hardware configurations. The purpose of this phase is not to single out one hardware configuration above all others but to determine which configurations can support the applications and if possible to rank these configurations on some sort of evaluation scale. Several steps are required to achieve the objectives of this phase and include:

1- a requirements study

2- a features specification study

3- an evaluation of qualified configurations

Once the first phase reaches step 3, the second phase of selection, which addresses the DMS, can begin.

The first two steps within this phase, the determination of the applications requirements and converting these into DMS requirements is probably the most significant. It is here that the test group must decide on the test criteria to be employed during the actual test. Once this decision is made, then the testing can begin.

Analysis was the first technique employed and, based on system documentation and user interviews, the most qualified systems for the applications in question were determined. Benchmarks then were run on all the DMSs to collect overall system performance data after which software monitors were employed to test specific DMS functions. The collected data was

then manually analyzed and used to select the best combination of hardware and data management system for the MIS.

SECTION VI

DMS EVALUATION DEVELOPMENT RECOMMENDATIONS

The intent of this paper has been to develop and present a methodology for the testing of data management systems. Section II identifies the various attributes that should comprise a DMS and summarizes the techniques that can be employed in implementing these capabilities. Section III, discusses the most common measurement techniques that can be used to measure the capabilities of the aforementioned attributes and proposes a test methodology to be employed in the testing of DMSs. Section IV attempts to draw a correlation between the attributes covered in Section II and the test techniques analyzed in Section III by pairing particular attributes with particular measurement techniques. Finally, Section V, through the utilization of scenarios, illustrates how the methodology (incorporating the test pairs concepts developed in Section IV) would be employed in the solution of some typical DMS measurement problems.

This section, now, will attempt to summarize the conclusions arrived at during the preceding sections and will propose some recommendations for the continued development of a DMS Test Methodology.

1. CONCLUSIONS

The measurement techniques that will be the most frequently used and therefore be of the most value during the implementation of the aforementioned DMS Test Methodology are analysis, benchmark programs and software monitors. It is anticipated that many if not most, testing problems will be solved by utilizing these three techniques in the above sequence. Analysis will filter out all but the most qualified systems (qualified, that is to the degree with which they satisfy user requirements). Benchmark programs, then, will be run to collect timing and performance data for the system as a whole. Finally, software monitors will provide the degree of definitiveness required to quantitatively measure actual DMS functions so that comparisons can be drawn between/among like systems.

The basis for this conclusion stems from the nature of the techniques. First, they possess the greatest degree of flexibility out of all the techniques considered and, secondly, on a cost/performance basis, they provide the most pertinent data for the least expenditure of time and money. Finally, both the benchmark programs and the software monitors, once generated, can be used again and again in the fine tuning of an operational system. For example, the software monitors can be periodically embedded in the DMS and the benchmark programs executed against the data base to measure any system degradation or improvement that may have occurred because of changes in file structure and/or size. New benchmarks also can be run against the monitored system to determine the DMS performance based on new applications.

Such techniques permit a data base administrator or manager to continually monitor system performance. Data base and/or application changes

that necessitate particular DMS changes (file structure, access methods and/or organization) would be promptly identified and an adequate amount of lead time to find and implement the changes required to avoid system degradation would be provided.

The recommendation of the above techniques is not meant to imply that the other techniques, numerical scoring, kernel analysis, hardware monitors, modeling and simulation would not be utilized in the testing and evaluation of data management systems, but that their usage would be infrequent and somewhat limited. For example, simulation, because of the cost and difficulty associated with its implementation would seldom be used. The time and effort required to simulate a complete DMS or simply one part of it normally would be prohibitive, therefore, other and more convenient techniques would be utilized. Modeling would be eliminated from consideration for the same reasons. Exceptions to this could occur when a system is to be selected for a multitude of users. Then, the costs involved in simulating or modeling a DMS may be justified because the model could be exercised by all the diverse applications to determine its acceptability to all users. This would certainly be more efficient than for each user to independently conduct acceptance tests on the selected DMS.

The utilization of pre-packaged software models, such as FORMS would also be infrequent because their range of capability is limited by the nature of the model. FORMS can be an invaluable tool in the testing of various file structures and access methods, but this is the extent of its capabilities. Other modeling/simulation packages which do not possess such a handicap usually suffer from a lack of specificity, and the collected data, because of this lack, is difficult to interpret and often inclusive.

Hardware monitors often will be eliminated as a DMS analysis and measurement tool because of their lack of availability, their rental or purchase cost and the ensuing data reduction problem that accompanies their use. They also lack the definitiveness to gather pertinent data regarding particular DMS functions, although they can be most useful in identifying overall system problems such as poor CPU-I/O overlap.

Kernel analysis, because it presumes the existence of some sort of software monitor or accounting system to collect the CPU times for various DMS functions, will be infrequently utilized unless such software packages already exist within the system. The only dissimilarity between kernel analysis and software monitors is the examination of the generated code that accompanies the use of kernel analysis. Therefore, unless some pre-existing software packages already monitor the desired DMS functions, test personnel would have to generate and embed software monitors within the DMS prior to conducting kernel analysis. It would be easier and more logical to simply utilize the software monitor technique.

Numerical scoring suffers from one glaring difficulty; that being the problem associated with the conversion of user applications requirements into specific DMS functional traits. This exercise is completely subjective and difficult to accomplish for even the most competent personnel. So many assumptions are required pertaining to the utilization of this technique that

the assignment of quantitative scores to analyzed DMSs is rather presumptuous. For example, the assignment of ratings both to particular DMS parameters and to the degree to which the attributes of a particular DMS conform to those parameters introduces such a degree of subjectivity into the measurement as to degrade the preciseness of the result.

The five aforementioned techniques, numerical scoring, hardware monitors, kernel analysis, modeling and simulation, although infrequently used, do rate inclusion within the DMS Test Methodology, because of their potential value in particular measurement and testing situations. The bulk of the measurement and testing however will be accomplished using analysis, benchmarks and software monitors.

2. RECOMMENDATIONS

The following recommendations result, first, from the problems encountered and secondly, from the conclusions derived during the compilation of the DMS Test Methodology. These recommendations neither attempt to solve all problems associated with the testing of data management systems, nor suggest that they will. If implemented, however, they will greatly advance the "state of the art" so that present and potential DMS users will be able to test a particular DMS or specific functions of same based on already available hard and concrete measurement data. The recommendations are as follows:

- o standardize the terminology regarding data management systems,
- o delineate the instrinsic characteristics of data management systems,
- o formalize methodology of specifying user requirements and converting them into DMS functional requirements,
- o support the design and development of machine independent data management systems,
- o investigate the possibility of using regression and cluster analysis to measure DMS performance,
- o design and develop standardized benchmark programs to measure the capabilities of present and potential data management systems,
- o identify the DMS functions that are appropriate candidates for software monitorization,
- o encourage DMS developers to include software monitors in the design of future systems and to embed them in the already operational systems, and
- o compile a compendium of standardized test results on all present data management systems and make this available on request to all present and potential DMS users.

These recommendations, if adopted, would constitute a giant step forward in the testing and evaluation of data management systems. What, at present, can best be described as a mercurial situation would be stabilized. Inefficient and redundant testing of the presently available DMSs could be

eliminated and potential and present DMS users could simply review the previously collected test results to determine which DMS would best fulfill their applications requirements. Some fine tuning of these results may be required if a particular user's applications requirements were rather unique, but much of the tedious analysis and measurement would have already been accomplished and from such an information base, efficient testing and selection could be accomplished. The following paragraphs describe each recommendation in detail and discuss the benefits that can be derived from their implementation.

a. Standardization of Terminology

The first step that must be taken regards the standardization of all terminology applicable to data management systems. Since one of the desired ends is a compendium of all DMS test results that would be available to all users, these very users first must speak the same language regarding data management systems. Otherwise confusion is bound to occur and the value of the compendium is degraded. The Data Base Group of the CODASYL Committee is presently engaged in this standardization effort. This is only part of the problem, however, for once standards are suggested by CODASYL, they still require adopting by all DMS users. If and/or when this is achieved, then the subsequent recommendations are much easier to implement.

b. Characteristic Delineation

A large number of software systems presently parade under the title of data management systems, however, the capabilities of each system are so diverse as to blur the definitiveness of the term. Once DMS terminology has been standardized, however, the minimum and maximum characteristics of a DMS can be more easily defined and the term DMS can assume a more precise meaning.

This definition will allow potential DMS users to face the test and evaluation process knowing full well that their initial list of candidates at least can perform the basic functions assumed to be part of all DMSs. This can greatly ease the cost and physical effort that would normally be applied to an initial analysis effort, because the initial review of all DMSs, to determine if they indeed do possess the general capabilities assumed by the user, would have already been accomplished.

This step would also aid in the establishment of general criteria against which newly emerging software systems can be judged to determine whether they should, in fact, be classified as data management systems.

c. Formalize Conversion of Application to DMS Requirements

This recommendation, perhaps, will be the most difficult to accomplish because of the multitude of variables that must be considered. There are, at present, a wide diversity of applications that require the services of a data management system. The process of relating specific DMS

functions (which are continually increasing) to their diverse applications becomes quite an effort when you consider the effect that these DMS functions have on the applications. A particular file structure and access method may "best" support a particular application, but what about other diverse tasks that require computer support within the operational environment? Is their execution going to be seriously degraded? Also, can the available physical storage handle the overhead associated with a particular file structure and access method? These are just some of the questions that arise when attempting to convert application to DMS requirements.

These problems, however, should not dissuade us from analyzing this topic to determine if a methodology can be developed to aid DMS analysts in the conversion of applications to DMS requirements. Such a methodology would be an invaluable tool in any DMS selection process, therefore such an effort should be given further consideration.

d. Machine-Independent Data Management Systems

Step 3 in the DMS Test Methodology illustrated the constriveness of host systems. At present, most systems are designed and developed to operate only on a particular hardware configuration (IDS and AIDS-HIS G-635, MADAPS - CDC 1604). This fact can result in the elimination of perhaps the most qualified system because of hardware incompatibility. Therefore, in order to widen the choices available to a potential DMS user, the design and development of machine independent data management systems should be supported.

Efforts in this regard are already in progress, for example, DM-1 and the Defense Intelligence Agency's Machine Independent Data Management System (MIDMS). These systems and systems like them will increase the options available to all installations involved in the testing and selection of a DMS which would provide a potential user with a better opportunity to select a system best suited to his needs.

e. Investigate the Utilization of Regression and Cluster Analysis as Data Analysis Techniques

Because of the speed, concurrency and volume of operations performed by a multi-programming third generation computer system, vast amounts of data will be collected by any active technique. This results in a data reduction and analysis problem of significant scope.

Regression and cluster analysis have been used with some success in the analysis of computer systems as a whole as illustrated in the Rand study Computer Performance Analysis: Applications of Accounting Data by R. Watson (2). It is recommended that these same techniques be studied to determine the feasibility of using them in the analysis of DMS test data. This would require the identification of those DMS and system factors that would be considered in the analysis, and the actual usage of these values against a known quantity to test the accuracy of the measurement.

Such techniques, if feasible, would greatly assist test personnel in the testing and subsequent evaluation of data management systems. For the resulting statistics would assume a higher level of validity and make the decision process a good deal easier.

f. Development of Standard Benchmark Programs

Standard benchmark programs should be developed to provide a consistent yardstick in the testing of all data management systems. These programs should be designed to test those functions that are found in every DMS; data description, file structuring and generation file maintenance and the data access manipulation and output capabilities. The benchmarks should be written in a higher level language to permit their execution within varied hardware environments and they must be designed to test specific aspects of the DMS to provide valid data on the systems stronger and weaker points.

These programs would be used to test all data management systems. Those systems presently in operation would be tested as would all new systems upon issue. The timing data collected during program execution would be compiled, and made available to all interested parties. The data would be ordered by type task performed (retrieval, update, generation, etc.) so that potential users would have a basis for comparing the available systems.

g. Identify DMS Functions to be Monitored

A list of DMS functions that are candidates for monitorization should be compiled to guide potential users of the software monitor technique. Such a compendium would eliminate the time now spent just in deciding which DMS attributes to select. The experience gained by those who have already embedded software monitors within DMSs, e.g., PRC/ISC's monitorization of MADAPS, should be compiled and disseminated to all potential DMS test personnel. Particular DMS system modules that, because of the programming techniques employed or the structure of the module itself, are poor candidates can be identified and thereby save other users a good deal of time and trouble. On the other hand, those particular DMS system modules that provide the most valuable statistics concerning system performance also can be indicated.

h. Design Software Monitor Embedded Data Management Systems

Rather than recommending, as the long term solution, the embedding of software monitors by potential users, DMS developers should be encouraged to design them into their systems when they are being developed. These monitors then could be used or not used based on the particular requirements of the user. It would be a relatively simple task to include them as part of the system design and yet it would save untold hours on the part of every potential DMS user. Additionally, the monitors could be embedded more efficiently within the DMS since it would be done during actual system coding rather than after the fact.

The developers of presently operational systems, also should be encouraged to modify their present systems to incorporate monitors within the appropriate modules.

The presence of such systems would permit system performance testing to be accomplished with relative ease. The only requirement levied on test personnel would be to initialize the monitorization process. The monitors also would be available for the purpose of "fine tuning" a system that has undergone numerous changes resulting from more and diverse applications, increased file volume, the addition of more files, etc. The monitors would be able to detect and identify any de adation of system performance due to these changes and provide a suft. ient amount of lead time to find and implement a solution.

An important aspect of software monitor development that requires mention here is the necessity to include data reduction, collation and analysis subroutines with the monitors. The collected data is useful only if it is in a format that is easily interpretable by the analyst, therefore, DMS developers must not neglect to incorporate such software packages within their systems.

i. Standard Test Results Compilation

All the effort expanded in accomplishing the previously defined recommendations would go for naught if the information is not collected and disseminated to all present and potential DMS users, therefore it is recommended that standard benchmarks be executed against all the software monitored data management systems and the results collected, compiled and disseminated. As more systems are tested using this procedure, the results would be appended to the compendium.

The end result would be a volume of DMS performance data that would be invaluable in any DMS selection process. Performance by application would be available and the capabilities of the various systems vis-a-vis particular applications could be easily determined. Even if the user feels his application(s) is unique, the compendium provides relevant timing and performance statistics for all systems, and the user can compare the performance of the various system modules against his own processing requirements.

Such a compendium would be beneficial not only from the standpoint of the potential user but also the system developer. Test results would be based on statistics derived from software monitors designed as part of the DMS (if the previous recommendation was implemented) rather than inefficient code segments implanted within the DMS by system users subsequent to its design and development. System users, therefore, would be able to view the DMS in its best light, and the capabilities of each system would not be tarnished due to poor generation and/or insertion of software monitors within the DMS.

The adoption of the nine previous recommendations would result in an important advancement regarding the testing and selection of data

management systems. Such software packages constitute a significant investment within any data processing environment, and therefore, the importance associated with selecting the right system cannot be overestimated. The methodology described herein provides present and potential system users with a tool to measure the various data management systems. This tool, however, can become expensive and difficult to employ depending on the complexity of the measurement problem. Benchmark programs may have to be written and perhaps software monitor code segments generated and embedded within the DMS. Such efforts would require the expenditure of a significant number of man hours particularly regarding the generation of software monitors. Additionally, many installations do not possess the expertise to embed monitors within a number of data management systems. Therefore, the aforementioned recommendations were made.

The two main aims of these recommendations are to standardize DMS testing and to eliminate the redundant and inefficient testing that is presently being performed. With the publication of standard test results, the need to actively test the candidate systems will be eliminated in most installations. Those possessing unique applications would be required only to generate the appropriate benchmarks and execute them against the various monitored systems; a much less costly exercise than starting from "scratch". The selection of a DMS, then, becomes a relatively analytical exercise, capable of being performed by a majority of users at a minimum cost.

REFERENCES

1. Walter, C.J., "Mathematical Techniques to Improve Hardware Accuracy of Graphic Display Devices", AFIPS, Vol. 3, SJCC, 1967, pp. 107-120.

   Williams, J.G., "Storage Utilization in a Memory Hierarchy When Storage Assignment is Performed by a Hashing Algorithm", CACM 14.3, March 1971, pp. 172-175.

2. Watson, R., Computer Performance Analysis: Applications of Accounting Data, RAND R-573-NASA/PR, May 1971.

3. Dowkont, Anthony J., William A. Morris and T. Dwight Buettell, A Methodology for Comparison of Generalized Data Management Systems: PEGS (Parametric Evaluation of Generalized Systems), Directorate of Computers, Electronic Systems Division Air Force Systems Command, United States Air Force, ESD-TR-67-2, March 1967.

BIBLIOGRAPHY

Abrahams, Paul W., Quantitative Methods of Information Processing System Evaluation, ITT Data and Information Systems Division, AD-433-220, October, 1963.

Alderige, Lamar, A User Oriented On-Line Data System, Syracuse University Research Corporation, AD-813-248, AD-813-082.

Apollo Systems Department, ADVISOR - Advanced Information System Operated Remotely, General Electric Company, 1968.

Aron, J. D., "Information Systems In Perspective," Computing Surveys, Vol. 1, No. 4, December, 1969, pp. 1-236.

Birly, Vorhaus, Weissman, A Study of Operating Systems for Data Management, SDC, AD-844-680, November, 1968.

Calingaert, Peter, "System Performance Evaluation: Survey and Appraisal," Communications of the ACM, Vol. 10, No. 1, January, 1967.

Canning, Richard G., "Data Management: File Organization," EDP Analyzer, Vol. 5, No. 12, December, 1967.

Cargaro, McCrea, Muhlhauser, Functional Specifications, DM-1, Auerbach Corporation, AD-855-587, November, 1968.

Cira, Paul F. and Edward L. Teger, Modifications to the On-Line Manipulation Techniques (MADAPS) System and Computer Program Descriptions, System Development Corporation, TM-WD-348/300700, April, 1971.

CODASYL Systems Committee, "Introduction to Feature Analysis of Generalized Data Base Management Systems," Communications of the ACM, Vol. 14, No. 5, May, 1971.

COMRESS, A Technical Description of SCERT.

Croke, J. J., Query Languages in Data Retrieval Systems: Final Report, MITRE Corporation, TM-3727, August, 1964.

Crowley, W., S. Forst, P. Harper, M. Rosenthal, and Dr. J. Sable, Reliability Central Automatic Data Processing Subsystem, RADC Research and Technology Division, Air Force Systems Command, RADC-TR-66-474, August, 1966.

Data Base Requirements Group, Data Base Management System Requirements, GUIDE-SHARE, November, 1970.

Preceding page blank

Dowkont, Anthony J., William A. Morris and T. Dwight Buettell, A Methodology for Comparison of Generalized Data Management Systems: PEGS (Parametric Evaluation of Generalized Systems), Directorate of Computers, Electronic Systems Division, Air Force Systems Command, United States Air Force, ESD-TR-67-2, March, 1967.

Fillat, Kraning, Generalized Organization of Large Data Bases, MIT(MAC), AD-711-060, June, 1970.

Flores, Ivan, Data Structure and Management, Prentice-Hall, Inc., 1970.

Fry, Bramson, Fried, Grabowsky, Jeffries, Mahle, Sternick, Data Management Systems Survey, MITRE Corporation, January, 1969.

Goldstein, Strnad, The MACAIMS Data Management System, MIT(MAC), AD-721-620, April, 1971.

Graham, File Management and Related Topics, MIT(MAC), AD-712-068, September, 1970.

Griffel, David M. and Stuart D. McIntosh, Large Disparate Data Bases, Center for International Studies, Massachusetts Institute of Technology, September, 1968.

Johnson, R. R., "Needed: A Measure for Measure", DATAMATION, December 15, 1970.

Kellog, CONVERSE: A System for the Online Description and Retrieval of Structured Data Using Natural Language, SDC, AD-654-622, May, 1967.

Kolence, Kenneth W., "A Software View of Measurement Tools," DATAMATION, January 1, 1972.

Loper, W. E., Application of Data Management Systems to Command Control, Naval Electronics Laboratory Center, Technical Document 112, April, 1971.

Lowe, Design Principles for an On-Line Information Retrieval System, Pennsylvania University, AFSOR 67-0423, December, 1966.

Lowe, Robert, On-Line Retrieval, Informatics, Inc., RADC TR69-304, TR70-44, November, 1969, April, 1970.

Madnick, Design Strategies for File Systems, MIT(MAC), AD-714-269, October, 1970.

Marcus, C. A., AIDS Users' Manual, Electronic Systems Division Air Force Systems Command, United States Air Force, F19(628)-71-C-0002, August, 1971.

Maynard, Denis, "Data Management," October, 1968.

McIntosh, Choate, Mittwede, Analysis of Major Computer Operating Systems, COMTRE Corporation, AD-715-919, August, 1970.

Merrow, Cady, On-Line File Manipulation Techniques for the Management Data Processing, SDC, RADC TR-69-362.

Merrow, Cady, On-Line File Manipulation Techniques, SDC, RADC TR-67-600.

Minker, Generalized Data Management Systems, Some Perspectives, University of Maryland, TR69-101, December, 1969.

Minker, Crook, Yeh, Analysis of Data Processing Systems, University of Maryland, TR69-99, December, 1969.

Minker, Sable, Relational Data System Study, Auerbach Corp., AD-720-263, September, 1970.

Muhlhauser, Data Management System, DM-1, Auerbach Corp., AD-844-695, November, 1968.

Mullins, Robert C., File Organization Modeling System User's Manual, PRC Information Sciences Company, F30602-71-C-0130, December, 1971.

NMCSSC, NIPS/360 Combat Analysis Data Management System, AD-878-491L, November, 1970.

Olle, T. William, "MIS: Data Bases," DATAMATION, November 15, 1970, pp. 47-50.

Schwartz, M. H., "MIS Planning," DATAMATION, September 1, 1970, pp. 28-30.

Senko, Lum, Palermo, Ghosh, Formatted File Organization Techniques, IBM Research Lab, RADC TR70-90, June, 1970.

Senko, Meadow, File Design - A Practical Approach, IBM, RADC TR-69-392, Vol. I, October, 1969.

Strnad, The Relational Approach to the Management of Data Bases, MIT(MAC), AD-721-619, April, 1971.

System Development Corporation, MADAPS Users Manual, TM-WD-348-400-00, July, 1971.

Walter, C. J., "Mathematical Techniques to Improve Hardware Accuracy of Graphic Display Devices," AFIPS, Vol. 3, SJCC, 1967, pp. 107-120.

Watson, R., Computer Performance Analysis: Applications of Accounting Data, RAND R-573-NASA/PR, May, 1971.

Welch, Bounds on Information Retrieval Efficiency in Static File Structures, MIT(MAC), June, 1971.

Williams, J. G., "Storage Utilization in a Memory Hierarchy When Storage Assignment is Performed in a Hashing Algorithm," CACM 14. 3, March, 1971, pp. 172-175.

Woolf, Ashby W., Analysis and Optimization of Multiprogrammed Computer Systems Using Storage Hierarchies, Michigan University, AD-730-325, August, 1971.

Weelley, G., Overview of the STIMS Modules, Documentation Incorporated, TM-007-1000-1.

Worldwide Military Command and Control System, Operational Capability Description for a Data Management System, SMC Project 41-67.

Ziehe, Data Management: A Comparison of System Features, TRACOR, Inc., AD-661-861, April, 1971.